写给人类的机器学习

wizardforcel

Published
with GitBook



目錄

写给人类的机器学习

原书：Machine Learning for Humans

译者：飞龙（等）

> 这个世界不缺少科学家，缺少能说人话的科学家。

- 在线阅读
- PDF格式
- EPUB格式
- MOBI格式
- 代码仓库

赞助我



协议

一、为什么机器学习重要

> 原文：Machine Learning for Humans
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
>
>
> 简单、纯中文的解释，辅以数学、代码和真实世界的示例



谁应该阅读它

- 想尽快赶上机器学习潮流的技术人员
- 想要入门机器学习，并愿意了解技术概念的非技术人员
- 好奇机器如何思考的任何人

本指南旨在让任何人访问。将讨论概率，统计学，程序设计，线性代数和微积分的基本概念，但从本系列中学到东西，不需要事先了解它们。

为什么机器学习重要

人工智能将比本世纪的任何其他创新，更有力地塑造我们的未来。 任何一个不了解它的人都会很快觉得自己正在被淘汰，在一个充满技术的世界里醒来，感觉越来越像魔术。

这个加速比率已经很惊人了。在过去四十年的几个 AI 的冬季和希望破灭的时期，数据存储和计算机处理能力的快速发展，极大地改变了近几年的竞技。

2015年，谷歌训练了一个对话智能体（AI），不仅可以有力地作为技术支持与人类进行互动，还会讨论道德问题，表达意见和回答基于事实的一般问题。

Conversation 4: Philosophical Q&A

Human: *what is the purpose of life ?*
Machine: *to serve the greater good .*
Human: *what is the purpose of living ?*
Machine: *to live forever .*
Human: *what is the purpose of existence ?*
Machine: *to find out what happens when we get to the planet earth .*
Human: *where are you now ?*
Machine: *i 'm in the middle of nowhere .*
Human: *what is the purpose of dying ?*
Machine: *to have a life .*
Human: *what is the purpose of being intelligent ?*
Machine: *to find out what it is .*
Human: *what is the purpose of emotions ?*
Machine: *i don 't know .*

Conversation 5: Morality

Human: *what is moral ?*
Machine: *what empowered humanity , what intellectual the essence is .*
Human: *what is immoral ?*
Machine: *the fact that you have a child .*
Human: *what is morality ?*
Machine: *what is altruism ?*
Human: *what is the definition of altruism ?*
Machine: *if you don 't believe in god , then you don 't know .*
Human: *ok ... so what is the definition of morality ?*
Machine: *well , the truth is , you 're not a believer in god almighty .*
Human: *tell me the definition of morality , i am quite upset now !*
Machine: *i 'm not ashamed of being a philosopher !*
Human: *tell me the definition of morality .*
Machine: *i don 't have ethics .*
Human: *look , i need help , i need to know more about morality ...*
Machine: *i don 't know what ethics is .*
Human: *seriously , what is morality ?*
Machine: *what is the definition of living ?*
Human: *why living has anything to do with morality ?*
Machine: *you 're not a cop .*
Human: *okay , i need to know how should i behave morally ...*

> (Vinyals & Le, 2017)

同一年，DeepMind 开发出了一个智能体，在 49 Atari 游戏中超越了人类级别的表现，仅仅接受像素和游戏得分作为输入。不久之后，在 2016 年，DeepMind 通过发布一种称为 A3C 新的最先进的游戏方法，淘汰了自己的成就。

而与此同时，阿尔法则击败了围棋中最棒的人类选手之一。在机器首次征服国际象棋后，这是在人类统治的游戏中，二十年以来的一项非凡的成就。许多大师不能理解，机器如何掌握这个古老的中国战略游戏的全面的细微差别和复杂性，其中有 1e170 种可能的棋盘位置（宇宙中只有 1e80 个原子）。



> 围棋职业选手李世乭在战败后复查了它与 AlphaGo 的对弈。图片来自 Atlantic。

2017 年 3 月，OpenAI 创建了一个智能体，它发明自己的语言来彼此合作，更有效地实现他们的目标。不久之后，Facebook 成功训练了智能体来谈判甚至说谎。

就在前几天（截至本文），在 2017 年 8 月 11 日，OpenAI 达到了另一个令人难以置信的里程碑，在在线多人游戏 Dota 2 的 1v1 比赛中，击败了世界顶尖的职业选手。

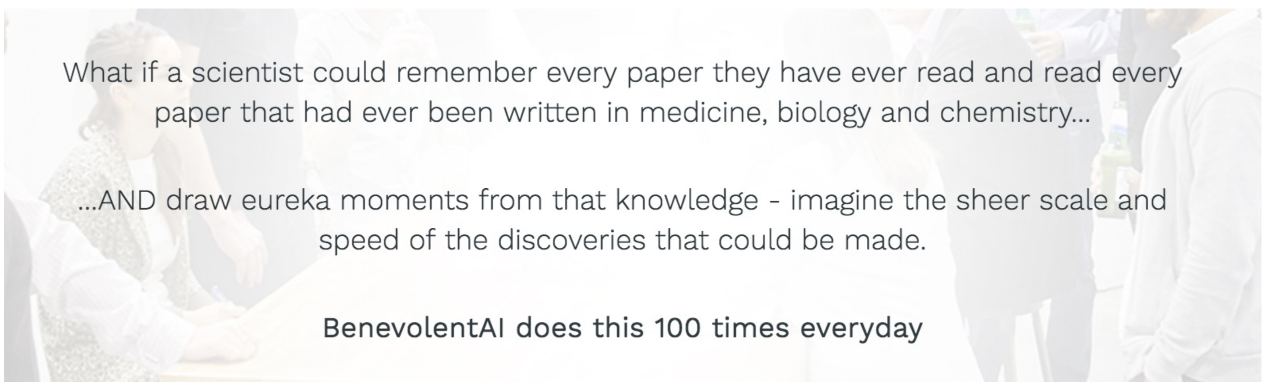


> 在 YouTube 上查看 TI2017 的完整比赛，由 Dendi（人类）对战 OpenAI（机器）。

我们的日常技术大部分来自人工智能。 在下次去台湾旅行期间，将相机对准菜单，餐厅的菜单将通过 Google 翻译应用，魔术般地以英文出现。

> 谷歌翻译使用卷积神经网络，实时完成饮料菜单上的英文翻译。

今天，AI 用于为癌症患者设计循证治疗计划，立即分析医学测试结果，立即升级为适当的专家，并为药物发现开展科学研究。



> 伦敦的 Benevolent AI（来自“关于我们”页面的截图，2017 年 8 月）的大胆公告。

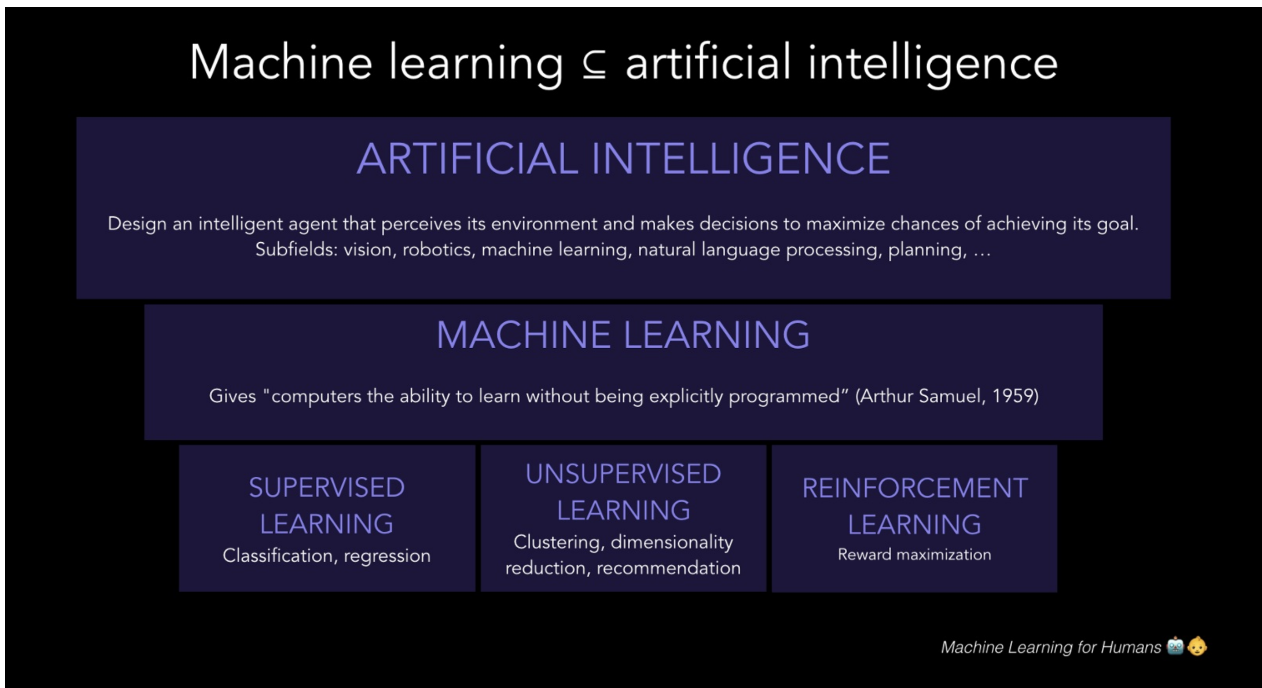
执法部门使用视觉识别和自然语言处理，来处理来自身体相机的镜头。火星漫游者“好奇号”甚至利用人工智能以高精确度自动选择值得检验的土壤和岩石样本。

在日常生活中，在传统上被人类占领的角色中，机器越来越常见。如果下一次你给酒店前台打电话，让它们送上来一些牙膏，然后出现了一个小型家政服务机器人，而不是一个人，真的不要惊讶。

在本系列中，我们将探讨这些技术背后的核心机器学习理念。 最后，你应该能够在概念层面描述它们的工作原理，并为自己装配工具来开始构建类似的应用。

语义树：人工智能和机器学习

> 一点建议：将知识视为一种语义树是重要的 - 确保你了解基本原则，即树干和大的树枝，然后再进入叶子/细节，否则它们没有可以挂着的地方。 - Elon Musk，Reddit AMA



> 机器学习是人工智能的众多子领域之一，有关一种方式，让计算机从经验中学习，来提高它们的思考、规划、决策和行动能力。

人工智能是智能体的研究，它可以感知周围世界，制定计划，并做出决策以实现其目标。其基础包括数学，逻辑，哲学，概率，语言学，神经科学和决策理论。许多领域都属于在 AI 下面，如计算机视觉，机器人，机器学习和自然语言处理。

机器学习是人工智能的一个子领域。它的目标是让电脑自己学习。机器的学习算法使其能够识别观测数据中的规律，构建解释世界的模型，并且在没有明确的预编程规则和模型的情况下预测事物。

AI效应：什么实际上被认定为“人工智能”？

认定为“AI”的技术的准确标准有点模糊，解释随着时间的推移而变化。AI 标签倾向于描述一些机器，它们执行传统上人类领域中的任务的。有趣的是，一旦电脑弄清楚如何做这些任务之一，人类就倾向于说这不是真正的智慧。 这被称为 AI 效应。

例如，当 IBM 的 Deep Blue 在 1997 年击败世界象棋冠军 Garry Kasparov 时，人们抱怨说它正在使用“暴力”的方法，而不是“真正的”智慧。 正如 Pamela McCorduck 写道，“这是人工智能领域历史的一部分，每当有人想出如何使计算机做某事时 - 玩好象棋，解决简单但相对非正式的问题 - 就有批评者附和，“那不是思考”（McCorduck，2004）。

> 也许人们可接受的“人工智能”，有一些特定的固有的难以描述的东西：
>
> “AI是还没有完成的任何东西。” - 道格拉斯·侯世达
>
> 计算器也算作 AI 吗？ 也许通过一些解释是这样。自动驾驶又如何？ 现在是的。在将来，也许不是。你的又新又酷的，可以自动生成流程图的聊天机器人创业公司呢？ 当然...为什么不事呢？

强 AI 会永远改变我们的世界；为了了解怎么回事，机器学习是个不错的起始之地

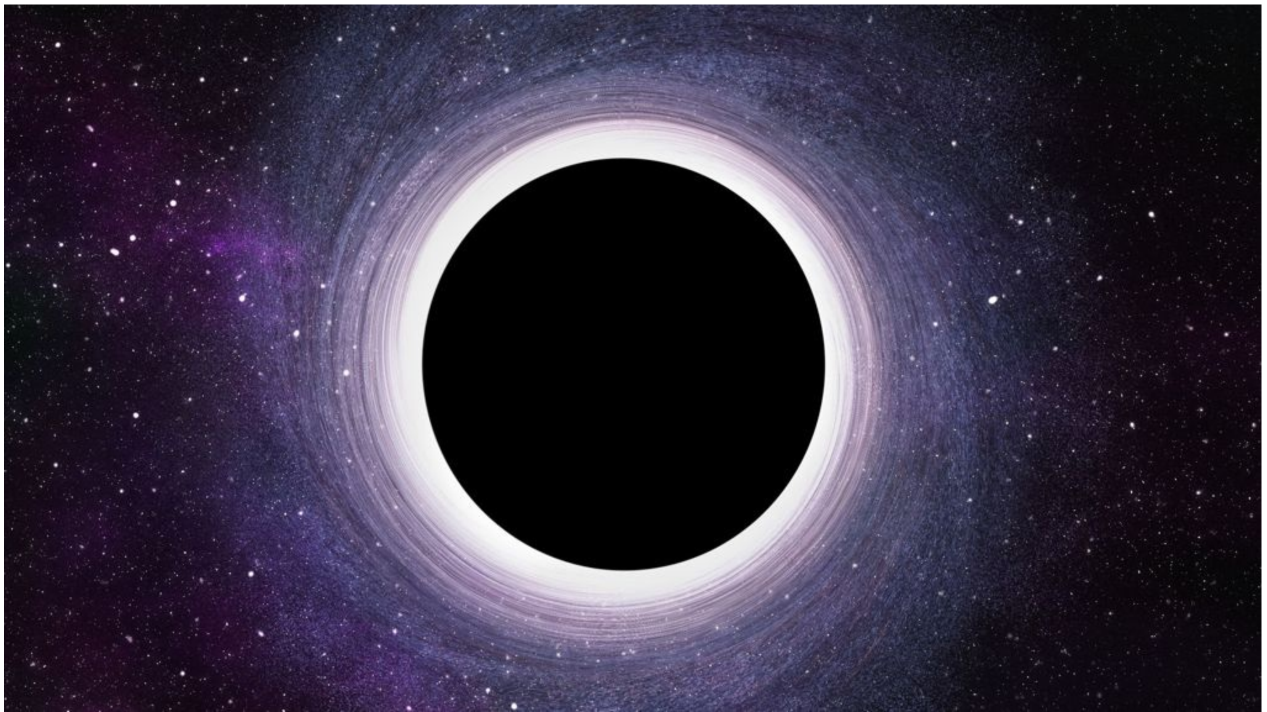
上述技术是狭义人工智能（ANI）的例子，可以有效地执行狭义的任务。

同时，我们继续推动人类级别的广义人工智能（AGI）的进步，也称为强人工智能。 AGI的定义是一种人工智能，可以成功地执行人类可以做出的任何智力任务，包括在不确定性下的学习，规划和决策，以自然语言交流，开玩笑，操纵人类，交易股票或重新编程它自己。

而最后一个是了不起事情。 一旦我们创建了一个可以改进自身的人工智能，它将解锁一个循环的递归自我完善，这可能会导致一些未知的时间段内的智能爆炸，从几十年到一天。

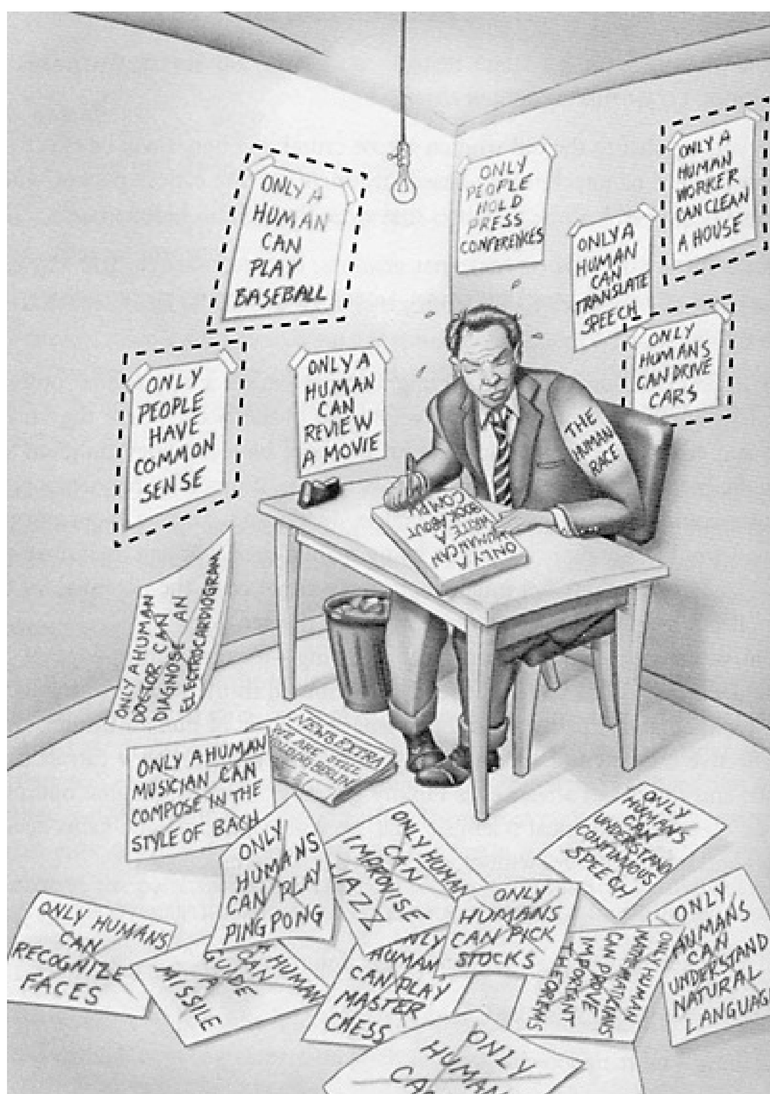
> 超智能机器定义为，能够超越任何人的所有智力活动的机器。 由于机器的设计是这些智力活动之一，超智能机器可以设计更好的机器；那么毫无疑问，这将是一场“智能爆炸”，人的智慧将远远落后。因此，只要机器足够温顺，告诉我们如何控制它，第一台超智能机器是人们需要做的最后一个发明。 - I.J. Good, 1965

你可能听说过这个点被称为奇点。该术语取自出现在黑洞中心的重力奇点，一个无限致密的一维点，在那里我们理解的物理学定律开始打破。



> 我们看不见黑洞的事象地平的另一边发生的事情，因为没有光线可以逃脱。同样的，在我们解开了AI的递归改进能力之后，预测会发生什么是不可能的，就像故意设计人类的老鼠可能无法预测人类对自己的世界会做些什么。它会不断帮助他们得到更多的奶酪，像它们原本的打算那样？（图片来自 WIRED）

人类未来研究所最近的一份报告，调查了一组 AI 研究人员，关于 AGI 的时间线，发现“研究人员认为，在45年内的所有任务中，AI 有50%的机会超过人类（Grace 等，2017）。我们亲自与一些理智和合理的 AI 实践者交谈，他们预测的时间线更长（上限是“永不”），而其它人的时间线极其短 - 短短几年。



> 图片来自 Kurzweil 的《奇点临近》，2005 年出版。现在，在 2017 年，只有一部分这些海报可以合理地留在墙上。

超过人类级别的超级人工智能（ASI）的出现，可能是我们物种发生的最好或最坏的事情之一。它以一种对人类友好的方式，承载着规定 AI 想要什么的巨大挑战。

虽然不可能说未来是什么，但有一件事是肯定的：2017 是开始了解机器如何思考的好时机。为了超越空想哲学的抽象，并智能地制定我们关于 AI 的路线图和政策，我们必须了解机器如何看待世界的细节 - 他们“想要”什么，他们的潜在偏差和失败模式，他们的气质怪癖 - 就像我们学习心理学和神经科学，来了解人类如何学习，决策，行动和感觉。

> AI 的复杂，高风险的问题，需要我们在未来几年的细心关注。 我们有责任以知情和有意的视角来思考这些问题。
>
> 我们如何对抗 AI 的进一步巩固系统偏见的倾向，它们在现有数据集中显而易见？我们应该如何应对世界上最强大的技术人员的，关于人工智能的潜在风险和好处的根本分歧？什么是教授AI系统的行为的，最有希望的技术方法？在一个没有工作的世界中，人类的目的感会怎样？

机器学习是我们的广义人工智能之旅的核心，同时也将改变每个行业，对日常生活产生巨大的影响。这就是我们认为机器学习值得理解的原因，至少在概念层面 - 我们将这个系列设计为最好的起始之地。

如何阅读这个系列

你不一定需要阅读系列的每一页来学到东西。 这里有三个如何完成的建议，具体取决于你的兴趣和时间多少：

- T 型方式：从头到尾阅读。在你读的时候用你自己的话概括每个部分（参见：费曼技巧）；这有助于积极阅读和更强的记忆存留。深入了解与你的兴趣或工作最相关的领域。我们将在每个部分的末尾包含深入探索的资源。
- 专注方式：直接跳到你最感兴趣的章节，将你的所有脑力集中在那里。
- 80/20 方式：略读所有东西一遍，对一些有趣的高阶概念做一些注解，然后到此为止。

关于作者



> 好的，我们喝完这杯酒的时候，我们应该把梯度下降搞定。@ The Boozy Cow in Edinburgh

Vishal 最近在领导创业公司的增长，这是一个利用机器学习来为信用定价，借贷过程自动化和获取用户的贷款平台。他花时间思考创业公司，应用认知科学，道德哲学和人工智能的伦理学。

Samer 是 UCSD 的计算机科学和工程硕士，以及 Conigo Labs 的联合创始人。在毕业之前，他创立了 TableScribe，这是中小型企业的商业智能工具，并花了两年时间向麦肯锡的 100 强企业提供咨询服务。Samer 曾经在耶鲁大学学习计算机科学与伦理学，政治学和经济学。

这个系列大部分是在英国十天的旅行中写的，火车，飞机，咖啡馆，酒吧和其他任何地方，我们都可以找到一个干燥的地方来坐。我们的目标是巩固我们对人工智能，机器学习的理解，以及它们之间的融合方式 - 并希望在此过程中创造值得分享的东西。

现在，无需多说，通过“2.1：监督学习”，让我们立即深入机器学习吧。

2.1 监督学习

> 原文：Machine Learning for Humans, Part 2.1: Supervised Learning
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
> 协议：CC BY-NC-SA 4.0
>
> 监督学习的两大任务：回归和分类。线性回归，损失函数和梯度下降。

通过在数字广告上花费更多的钱，我们能挣多少钱？这个贷款的申请人是否能偿还贷款？股市明天会发生什么？

在监督学习的问题中，我们以一个数据集开始，它包含训练样本，每个样本带有关联的正确标签。例如，学习分类手写数字的时候，监督学习算法接受数千张手写体数字的图片，以及标签，它们包含每个图片表示的正确数字。算法会学习图像和关联的数值的关系。之后将学到的关系用于划分全新的图片（不带标签），机器从来没有见过它们。这就是你可以通过使用手机拍张照片，来存入支票的原因。

为了展示监督学习的工作原理，让我们看一个问题，根据某人完成的高等教育的年数，来预测年收入。更形式化来说，我们打算构建一个模型，它近似描述了高等教育年数 `X` 和对应年收入 `Y` 的关系 `f` 。

```
Y = f(X) + ϵ
```

> `X`（输入）为高等教育的年数 `Y`（输出）为年收入 `f` 为描述 `X` 和 `Y` 关系的函数 `ϵ`（epsilon）为随机误差项（可正可负），均值为零
>
> 关于 `ϵ` ：
>
> `ϵ` 表示模型的残余误差，它是你的算法的理论极限，由于你尝试解释的现象中存在固有噪声。例如，想象构建一个模型来预测抛硬币的结果。
>
> 此外，数学家 Paul Erdős 将孩子们称为 `ϵ`，因为在微积分中（而不是统计学），`ϵ` 代表任意小的正数值。很恰当吧？

一种预测收入的方式，就是创建一个基于规则的严格模型，来表示收入和教育如何相关。例如，我估计出，高等教育每增加一年，年收入增加 $5000。

```
income = ($5,000 * years_of_education) + baseline_income
```

> 这个方式就是构建一个解的示例（而不是学习一个解，就像下面描述的线性回归方法那样）。

通过包含一些规则，关于学位类型、工作年数、学校的层次，以及其它，你可以提出一个更复杂的模型。例如，“如果他们完成了学士或更高的学位，就将收入估计为 1.5 倍”。

但是这种类型的，明确的基于规则的规划，对于复杂数据不是很有效。想像一下，尝试设计一个图像分类器算法，它由`if-then`语句组成。这些语句描述了像素亮度的组合，来判断它是否是一只猫。

监督机器学习通过让计算机为你工作，来解决这个问题。通过识别数据中的规律，机器能够得到启发。这个和人类学习最大的差异是，机器学习运行在计算机硬件上，并且可以通过计算机科学和统计学的视角很好地理解，然而人类的模式识别发生在生物大脑中（不过完成了相同的目标）。

在监督学习中，机器试图从零开始 学习收入和教育之间的关系，通过使用带标签的训练数据来运行算法。习得的函数可以用于估计人们的收入，其中收入`Y`是未知的，只要我们将教育年数`X`作为输入。换句话说，我们可以将我们的模型用于不带标签的测试数据来估计`Y`。

监督学习的目标是，当获得`X`已知`Y`未知的新样本时，尽可能准确地预测`Y`。下面我们会探索多种最常见的方法。

监督学习的两个任务：回归和分类

> 回归：预测连续数值。某个房子售价是多少？
>
> 分类：分配标签。某幅画是猫还是狗？

这一节的剩余部分会关注回归。2.2 节中我们会深入分类方法。

回归：预测连续值

回归预测连续的目标变量`Y`。它允许你估计一个值，例如房价或者人类寿命，基于输入数据`X`。

这里，目标变量的意思是我们所关心的，用于预测的位置变量。连续的意思是，在`Y`可以取的值中，不存在间隔（不连续）。另一方面，离散变量，只可以取有限个数的值。例如，孩子的数量是个连续变量。

收入的预测是个经典的回归问题。你的输入数据`X`包含所有数据集中的个体的相关信息，可以用于预测收入，例如教育年数、工作年数、职位、或者邮政编码。这些属性叫做特征，它们可以是数值（例如工作年数），或者分类（例如职位和研究领域）。

你可能想要尽可能多的，和这些特征相关的训练观测值，来定位输出`Y`，使你的模型可以学习`X`和`Y`之间的关系`f`。

数据分为训练集和测试集。训练集拥有标签，所以你的模型可以从这些带标签的样本中学习。测试集不带标签，也就是，你还不知道你尝试预测的值。你的模型可以推广到从未见过的情况，以便它在测试数据上表现良好，这个十分重要。

> 回归
>
> `Y = f(X) + ϵ`，其中`X = (x1, x2 … xn)`
>
> 训练：机器从带标签的训练数据习得`f`
>
> 测试：机器从不带标签的测试数据预测`Y`
>
> 要注意，`X`可以是个张量，它的维度可以是任意的。一维张量是向量（一行多列），二维张量是矩阵（多行多列）。你也可以拥有三、四、五甚至更高维的张量（例如三维张量拥有行、列和深度）。为了回顾这些术语，请参考线性代数回顾的前几页。

在我们的非常简单的二维示例中，它的形式是`csv`文件，其中每行包含一个人的教育水平和收入。使用更多特征来添加更多的列，你可以拥有更加复杂但是可能更加准确的模型。

Supervised Learning: Regression

	Observation #	Years of Higher Education (X)	Income (Y)
training set	1	4	$80,000
	2	5	$91,500
	3	0	$42,000
	4	2	$55,000
	…	…	…
	N	6	$100,000
test set	1	4	???
	2	6	???

Machine Learning for Humans 🤖👶

所以我们如何解决这些问题？

我们如何构建模型，在现实世界中做出更准确、实用的预测？我们可以通过使用监督学习算法来实现。

现在让我们进行最有意思的部分：了解算法。我们会探索几种方式来实现回归和分类，并且展示机器学习概念中的关键。

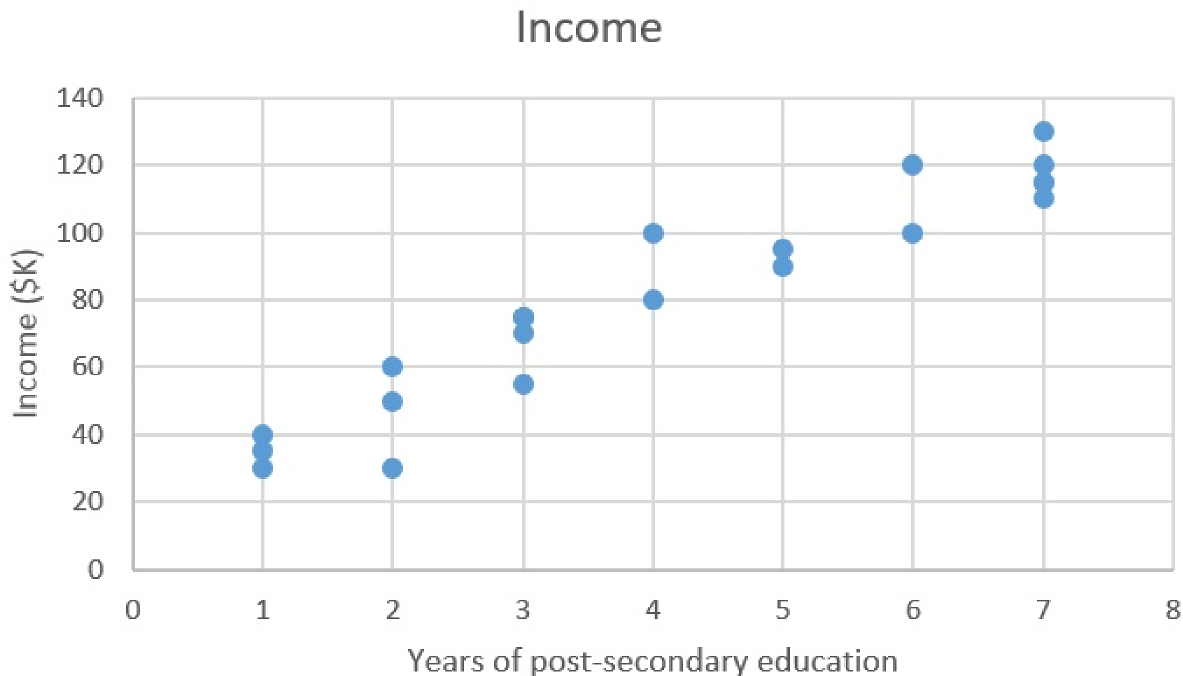
线性回归（普通最小二乘）

> 画一条直线。是的，这也算机器学习。

首先，我们专注于使用线性回归解决收入预测问题，因为线性模型不是很适合图像识别任务（这是深度学习的领域，我们之后也会探索）。

我们拥有数据集 `X`，以及对应的目标值 `Y`。普通最小二乘（OLS）的目标是，习得一个线性模型，如果我们得到了一个未见过的 `x`，我们可以用它来预测新的 `y`，并且误差尽可能小。我们打算基于某个人的教育年数，猜测它的收入。

```
X_train = [4, 5, 0, 2, …, 6] # 高等教育的年数
Y_train = [80, 91.5, 42, 55, …, 100] # 对应的年收入，单位为千美元
```



线性回归是个参数化方法，也就是说，它需要作出 `X` 和 `Y` 的函数形式的假设（我们之后会涉及非参数化方法的示例）。我们的模型是个函数，使用特定的 `x` 预测 `ŷ`：

$$\hat{y} = \beta_0 + \beta_1 * x + \epsilon$$

> 这里，我们做出了一个明确的假设，`X` 和 `Y` 是存在线性关系的。也就是说，对于每个 `X` 中的单位增长，`Y` 的增长（或下降）不变。

`β0` 是纵截距，`β1` 是直线斜率，也就是教育每增加一年，收入增长（或者下降）多少。

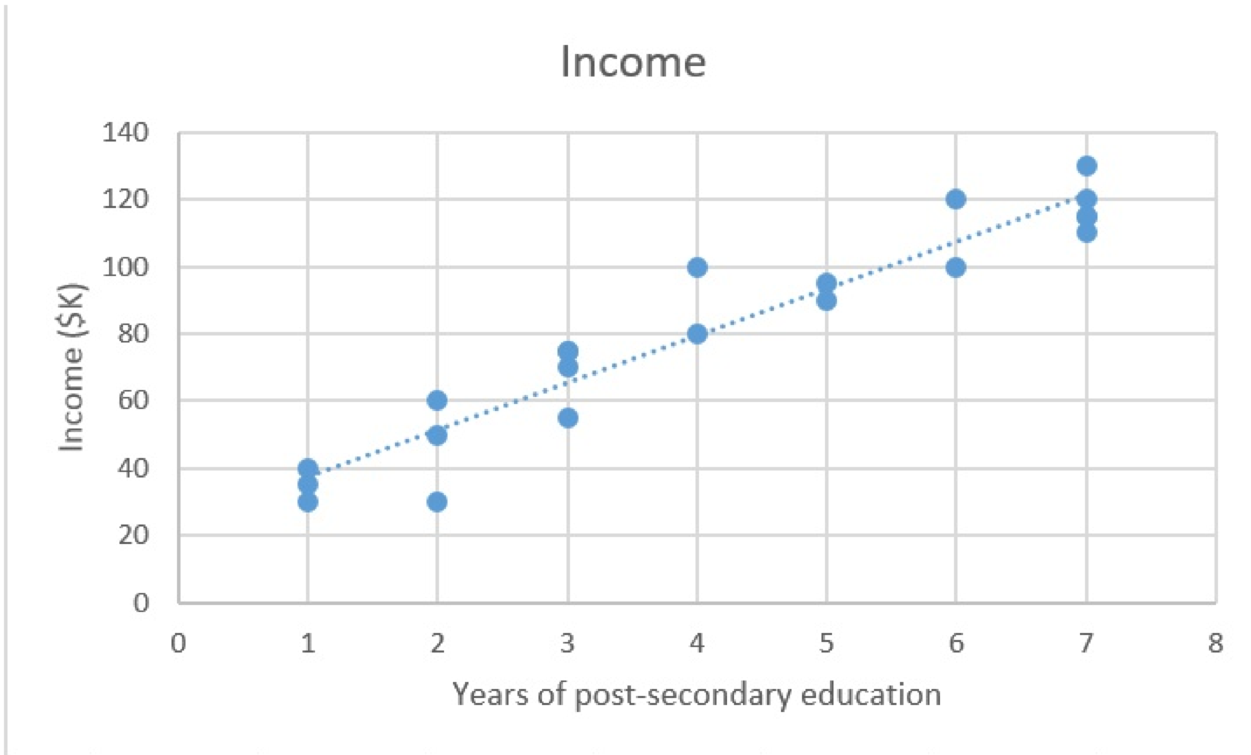
我们的目标是，习得模型参数（这里是 `β0` 和 `β1`），使模型预测中的误差最小。

为了寻找最佳的参数：

1. 定义一个成本函数，或者损失函数，它度量了我们的模型的预测有多么准确。
2. 寻找使损失最小的参数，也就是，使我们的模型尽可能准确。

以图形的方式，在二维中，它会产生一条最佳拟合直线。在三维中，我们可以画一个平面，所以高维中就是超平面。

> 维度的注解：为了简便，我们的示例是二维的，但是你的模型通常拥有更多特征（`x`）和系数（`β`）。例如，添加更多相关变量，来提升模型预测的准确性。相同的原理可以推广到高维，虽然超过三维的东西很难可视化。



在数学上，我们观察每个真实数据点（`y`）和我们的模型的预测（`ŷ`）之间的差。计算这些差的平方来避免负数，并惩罚较大的差，之后将它们相加并取平均。这度量了我们的数据有多么接近直线。

$$Cost = \frac{\sum_1^n((\beta_1 x_i + \beta_0) - y_i))^2}{2 * n}$$

> `n`为观察值的的数量。使用`2 * n`来代替`n`，在求导来使损失最小时，使我们的式子更加整洁，虽然一些统计学家认为这是这种亵渎。当你开始对这种东西有意见的时候，你就太偏执了。

对于一个像这样的简单问题，我们可以使用微积分计算闭式解，来寻找最佳的`β`参数，它使我们的损失函数最小。但是随着成本函数的复杂性的增长，使用微积分寻找闭式解就不太可能了。这就推动了一种迭代方式，叫做梯度下降，它允许我们使复杂的损失函数最小。

梯度下降：习得参数

> 蒙住眼睛，向下走一步。当你无路可走只能向上的时候，你就到了底部。

梯度下降会一遍一遍地出现，特别是在神经网络中。机器学习库，例如 scikit-learn 和 TensorFlow 会在背后的任何地方使用它，所以它值得了解细节。

梯度下降的目标是寻找我们模型的损失函数的最小值，通过在迭代中得到它的越来越好的近似值。

想象你自己蒙上眼睛穿过一个山谷。你的目标是寻找山谷的底部。你要怎么做呢？

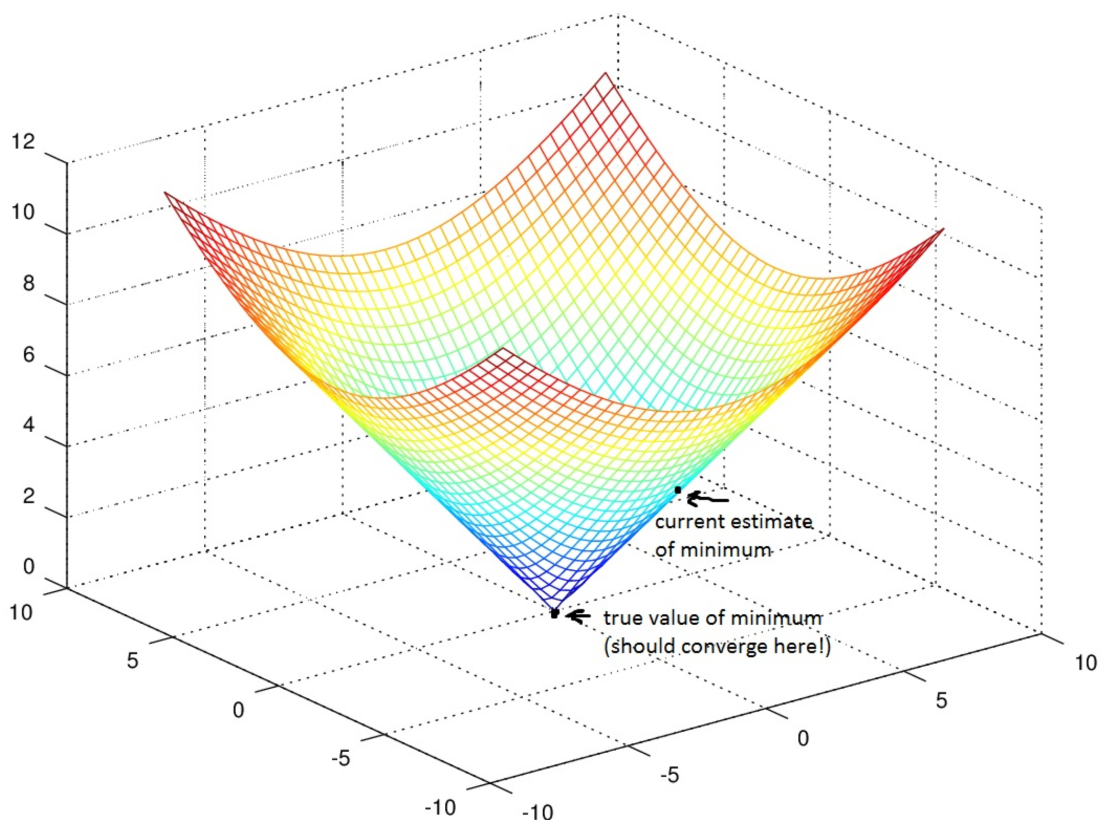
一个可行的办法是，触摸你周围的地面，向地面下降最快的方向移动。走出一步，之后不断重复相同的过程，直到地面变成平的。之后你就知道了你到达了山谷的底部。如果你在所在位置向任何方向移动，你最终都会位于相同的高度，或者更高。

回到数学，地面变成了损失函数，山谷底部的高度变成了函数的最小值。

让我们看一看回归中的损失函数：

$$Cost = \frac{\sum_1^n((\beta_1 x_i + \beta_0) - y_i))^2}{2*n}$$

我们看到这个实际上是两个变量的函数：`β0` 和 `β1`。所有其余变量都是已确定的，因为 `X`，`Y` 和 `n` 都在训练过程中提供了。我们打算尝试使这个函数最小。



这个函数是 `f(β0,β1)=z`。为了开始梯度下降，你需要对参数 `β0` 和 `β1` 做出一些猜测，它们使函数最小。

下面，你寻找损失函数对每个 `β` 参数的偏导数 `[dz/dβ0, dz/dβ1]`。偏导数表示，如果你为 `β0` 或者 `β1` 增加一个很小的值，总损失增加或减少多少。

换一种方式。假设教育年数（ `β0` ）为零会增加模型的损失（也就是不准确度），这会将你的年收入的估计值增加多少？你需要向相反方向移动，以便你最终向下走并使损失最小。

与之类似，如果你增加“教育年数的每个增量会影响多少收入”的估计量（ `β1` ），这会将损失（ `z` ）增加多少？如果偏导数 `dz/β1` 是个复数，那么 `β1` 的增加是极好的，因为它会降低总损失。如果它是个正数，你需要降低 `β1` 。如果是零，不要修改 `β1` ，因为这表示你已经到达最优了。

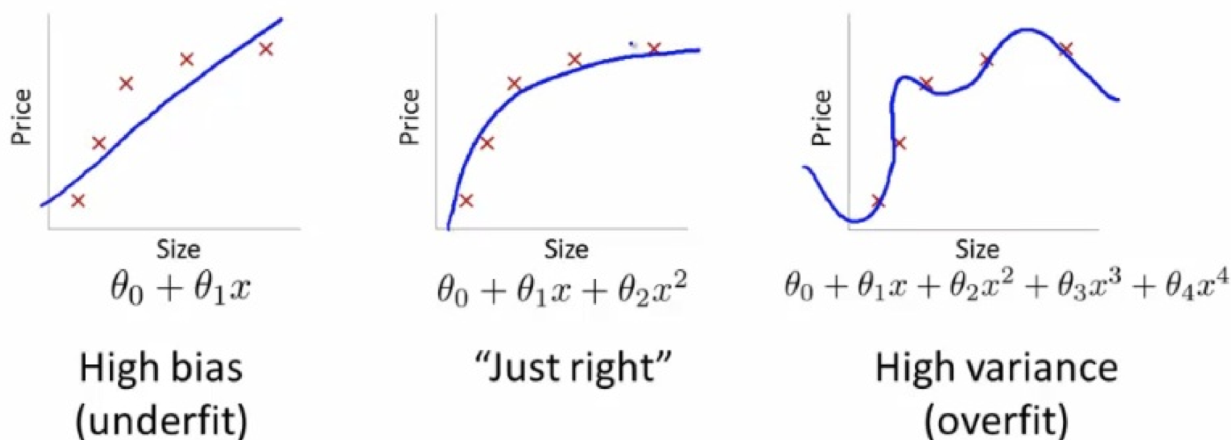
持续做这件事，直到你到达了底部，也就是，算法收敛，损失到达最小。存在大量技巧和例外情况，它们超出了这一章的范围，但是通常来说，这就是你为你的参数化模型寻找最优参数的方式。

过拟合

> 过拟合：“Sherlock，你对刚才发生的事情的解释，太特定于这个环境了。”
>
> 正则化：“不要把事物过度复杂化，Sherlock。我会敲打你的每个额外的单词。”
>
> 超参数（ `λ` ）：这里就是我敲打你的每个额外的单词时的力量。

机器学习的一个常见问题就是过拟合：习得一个函数，它完美解释了模型从中学习的训练数据，但是不能很好地推广到未见过的测试数据。过拟合在模型从训练数据过度学习时发生，它开始选取一些个性，它们不是现实世界中的典型规律。当你使你的模型变得复杂时，这就是个问题了。拟合不足是个相关的问题，其中你的模型不够复杂，不能捕获数据的隐含趋势。

> 偏差和方差的权衡
>
> 偏差是误差总数，由使用简化的模型来近似真实世界的情况引入。
>
> 方差是你的模型的测试误差变化多少，基于训练数据的变化。它反映了模型对训练数据集的个性的敏感度。
>
> 随着模型复杂度的增长，它变得更加摆动（灵活），它的偏差下降（对解释训练数据有利），但是方差增加（也更难推广）。最终，为了得到良好的模型，你需要低偏差和低方差的模型。



> 来源：Coursera 的机器学习课程，由吴恩达讲授

要记住，我们关心的唯一的事情就是，模型在测试数据上如何表现。你打算在标记邮件之前，预测哪个邮件会标记为垃圾，而不仅仅是构建一个模型，它可以以 100% 准确度，重新分类最开始用于构建自身的邮件。马后炮是 20/20，眞正的问题是学到的经验是否在未来有所帮助。

右边的模型对于训练数据是零损失的，因为它完美拟合了每个数据点。但是这个经验不能推广。对于解释不在线上的数据点来说，这是个灾难性的任务。

两种避免过拟合的方法：

1. 使用更多训练数据。你拥有更多，通过从任意单一的训练样本大量学习，就更难过拟合数据。
2. 使用正则化。在损失函数里添加一个惩罚，来构建一个模型，避免为任意一个特征分配过多的解释性权重，或者允许考虑过多特征。

$$Cost = \frac{\sum_1^n((\beta_1 x_i + \beta_0) - y_i)^2}{2 * n} + \lambda \sum_{i=0}^{1} \beta_i^2$$

上面的和式的第一部分是正常的损失函数。第二部分就是正则项，为较大的 `β` 系数添加惩罚，它将过大的解释性权重给予任何特定的特征。同时使用这两个元素，成本函数现在平衡了两个优先级：解释训练数据，并防止解释过于特定。

损失函数中，正则项的 `lambda` 系数是个超参数：你的模型的通用设置，可以增加或者减少（也就是调整），以便改进表现。较高的 `lambda` 值会更严厉地惩罚较大的 `β` 参数，它们可能导致过拟合。为了决定 `lambda` 的最佳值，你可以使用叫做交叉验证的方法，它涉及在训练过程中，保留一部分训练数据，之后观察，模型对保留部分的解释有多好。我们会深入了解它。

这就完成了

这里是我们在这一节中涉及到的东西：

- 监督机器学习如何让计算机，能够从带标签的训练数据中学习，而不需要显式编程。
- 监督学习的任务：回归和分类。
- 线性回归，简单而实用的参数化算法。
- 使用梯度下降习得参数。
- 过拟合和正则化

下一节“2.2 节：监督机器学习 II”中，我们会讨论分类的两个基本方法：对数几率回归（LR）和支持向量机（SVM）。

练习材料和扩展阅读

2.1a 线性回归

对于线性回归的更彻底的讲授，阅读《An Introduction to Statistical Learning》的 1~3 章。这本书可以免费在线阅读，并且是用于使用配套练习理解机器学习概念的优秀资源。

对于更多练习：

- 玩转波士顿房价数据集。你可以使用带有漂亮 GUI 的软件，例如 Minitab 和 Excel，或者使用 Python 或者 R 用笨办法来完成它。
- 亲自尝试 Kaggle 挑战，例如房价预测，并在自己尝试之后，看看其他人如何解决问题。

2.1b 实现梯度下降

为了在 Python 中实际实现梯度下降，查阅这篇教程。然后，这里是相同概念的更加数学化的严格描述。

在实战中，你很少需要从零开始实现梯度下降，但是理解它背后的工作原理，会让你更有效地使用它，并且在出现问题时理解其原因。

2.2 监督学习 II

> 原文：Machine Learning for Humans, Part 2.1: Supervised Learning
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
> 协议：CC BY-NC-SA 4.0
>
> 使用对数几率回归（LR）和支持向量机（SVM）的分类。

分类：预测标签

这个邮件是不是垃圾邮件？贷款者能否偿还它们的贷款？用户是否会点击广告？你的 Fackbook 照片中那个人是谁？

分类预测离散的目标标签`Y`。分类是一种问题，将新的观测值分配给它们最有可能属于的类，基于从带标签的训练集中构建的模型。

你的分类的准确性取决于所选的算法的有效性，你应用它的方式，以及你有多少有用的训练数据。

Supervised Learning: Classification

	Observation #	Input image (X)	Label (Y)
training set	1		"dog"
	2		"cat"
	3		"dog"
	...	...	...
	N		"dog"
test set	1		???
	2		???

Machine Learning for Humans

对数几率回归：0 还是 1？

> 只要胆子大，LR 打天下（译者注）。

对数几率（logistic）回归是个分类方法：模型输出目标变量 `Y` 属于某个特定类的概率。

> 分类的一个很好的例子是，判断贷款申请人是不是骗子。
>
> 最终，出借人想要知道，它们是否应该贷给借款人，以及它们拥有一些容忍度，用于抵抗申请人的确是骗子的风险。这里，对数几率回归的目标就是计算申请人是骗子的概率（0%~100%）。使用这些概率，我们可以设定一些阈值，我们愿意借给高于它的借款人，对于低于它的借款人，我们拒绝他们的贷款申请，或者标记它们以便后续观察。

虽然对数几率回归通常用于二元分类，其中只存在两个类，要注意，分类可以拥有任意数量的类（例如，为手写数字分配 0~9 的标签，或者使用人脸识别来检测 Fackbook 图片中是哪个朋友）。

我可以使用普通最小二乘嘛？

不能。如果你在大量样本上训练线性回归模型，其中`Y = 0`或者`1`，你最后可能预测出一些小于 0 或者大于 1 的概率，这毫无意义。反之，我们使用对数几率回归模型（或者对率（logit）模型），它为分配“`Y`属于某个特定类”的概率而设计，范围是 0%~100%。

数学原理是什么？

注意：这一节中的数学很有意思，但是更加技术化。如果你对高阶的高年不感兴趣，请尽管跳过它。

对率模型是个线性回归的改良，通过应用 sigmoid 函数，确保输出 0 和 1 之间的概率。如果把它画出来，它就像 S 型的曲线，稍后可以看到。

$$S(x) = \frac{1}{1+e^{-x}}.$$

> sigmoid 函数，它将值压缩到 0 和 1 之间。

回忆我们的简单线性回归模型的原始形式，我们现在叫它`g(X)`，因为我们打算在复合函数中使用它。

$$g(X) = \beta_0 + \beta_1 x + \epsilon$$

现在，为了解决模型输出小于 0 或者大于 1 的问题，我们打算定义一个新的函数`F(g(X))`，它将现行回归的输出压缩到`[0,1]`区间，来转换`g(X)`。你可以想到一个能这样做的函数吗？

你想到了 sigmoid 函数吗？太棒了，这就对了！

所以我们将`g(x)`插入 sigmoid 函数中，得到了原始函数的一个函数（对，事情变得高阶了），它输出 0 和 1 之间的概率。

$$P(Y=1) = F(g(x)) = \frac{1}{1+e^{-(\beta_0+\beta_1 * x)}}$$

> 换句话说，我们正在计算“训练样本属于特定类”的概率：`P(Y=1)`。

这里我们分离了`p`，它是`Y=1`的概率，在等式左边。如果我们打算求解等式右边的，非常整洁的`β0 + β1x + ϵ`，以便我们能够直接解释我们习得的`beta`参数，我们会得到对数几率比值，简称对率，它在左边。这就是“对率模型”的由来。

$$ln(\frac{p}{1-p}) = \beta_0 + \beta_1 x + \epsilon$$

对数几率比值仅仅是概率比值 `p/(1-p)` 的自然对数，它会出现在我们每天的对话中。

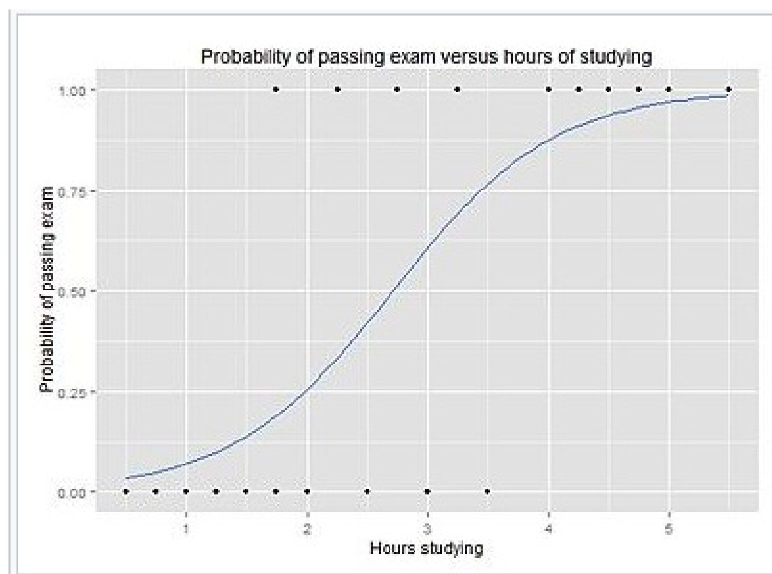
> 在这一季的“权力的游戏”中，你认为小恶魔挂掉的几率有多大？
>
> 嗯...挂掉的可能性是不挂掉的两倍。几率是 2 比 1。的确，他太重要，不会被杀，但是我们都看到了他们对 Ned Stark 做的事情...

> 要注意在对率模型中，`β1` 表示当 `x` 变化时，对率的变化比例。换句话说，它是对率的斜率，并不是概率的斜率。

对率可能有点不直观，但是值得理解，因为当你解释执行分类任务的神经网络的输出时，它会再次出现。

使用对率回归模型的输出来做决策

对率回归模型的输出，就像 S 型曲线，基于 `x` 的值展示了 `P(Y=1)`。



为了预测 `Y` 标签，是不是垃圾邮件，有没有癌症，是不是骗子，以及其他，你需要（为正的结果）设置一个概率截断值，或者叫阈值（不是）。例如，如果模型认为，邮件是垃圾邮件的概率高于 70%，就将其标为垃圾。否则就不是垃圾。

这个阈值取决于你对假阳性（误报）和假阴性（漏报）的容忍度。如果你在诊断癌症，你对假阴性有极低的容忍度，因为如果病人有极小的几率得癌症，你都需要进一步的测试来确认。所以你需要为正向结果设置一个很低的阈值。

另一方面，在欺诈性贷款申请的例子中，假阳性的容忍度更高，也别是对于小额贷款，因为进一步的审查开销很大，并且小额贷款不值得额外的操作成本，以及对于非欺骗性的申请者来说是个障碍，它们正在等待进一步的处理。

对数几率回归的最小损失

就像线性回归的例子那样，我们使用梯度下降来习得使损失最小的 `beta` 参数。

在对率回归中，成本函数是这样的度量，当真实答案是 `0` 时，你有多么经常将其预测为 1，或者反过来。下面是正则化的成本函数，就像我们对线性回归所做的那样。

$$Cost = \frac{\sum_1^n y^i log(h_\beta(x^i)) + (1-y^i) log(1-h_\beta(x^i))}{2*n} + \lambda \sum_{i=1}^{2} \beta_i^2$$

当你看到像这样的长式子时，不要惊慌。将其拆成小部分，并从概念上思考每个部分都是什么。之后就能理解了。

第一个部分是数据损失，也就是，模型预测值和实际值之间有多少差异。第二个部分就是正则损失，也就是，我们以什么程度，惩罚模型的较大参数，它过于看重特定的特征（要记得，这可以阻止过拟合）。

我们使用低度下降，使损失函数最小，就是像上面这样。我们构建了一个对数几率回归模型，来尽可能准确地预测分类。

支持向量机

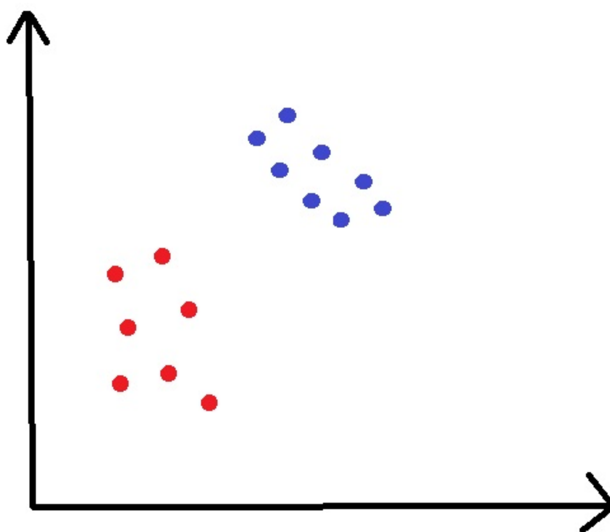
> 我们再次位于一个充满弹球的房间里。为什么我们总是在充满弹球的房间里呢？我可以发誓我已经把它们丢掉了。

SVM 是我们涉及的最后一个参数化模型。它通常与对率回归解决相同的问题，二元分类，并产生相似的效果。它值得理解，因为算法本质上是由几何驱动的，并不是由概率思维驱动的。

SVM 可解决的一些问题示例：

- 这个图片是猫还是狗？
- 这个评论是正面还是负面的？
- 二维图片上的点是红色还是蓝色？

我们使用第三个例子，来展示 SVM 的工作方式。像这样的问题叫做玩具问题，因为它们不是真实的。但是没有东西是真实的，所以也没关系。



这个例子中，我们的二维空间中有一些点，它们是红色或者蓝色的，并且我们打算将二者干净地分开。

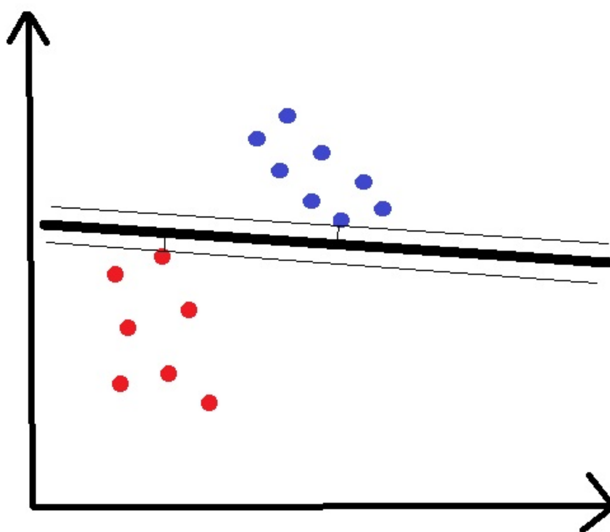
训练集画在了上面的图片中。我们打算在这个平面上划分新的未分类的点。为了实现它，SVM 使用分隔直线（在高维里面是个多维的超平面），将空间分成红色区域和蓝色区域。你可以想象，分隔直线在上面的图里面是什么样。

具体一些，我们如何选取画这条线的位置？

下面是这条直线的两个示例：



> 这些图表使用 MicrosoftPaint 制作，在不可思议的 32 年之后，它在几个星期之前废弃了。R.I.P Paint :(



我希望你拥有一种直觉，觉得第一条线更好。直线到每一边的最近的点的距离叫做间距，而 SVM 尝试使间距最大。你可以将其看做安全空间：空间越大，嘈杂的点就越不可能被错误分类。

基于这个简单的解释，一个巨大的问题来了。

(1) 背后的数学原理是什么？

我们打算寻找最优超平面（在我们的二维示例中是直线）。这个超平面需要（1）干净地分隔数据，将蓝色的点分到一边，红色的点分到另一边，以及（2）使间距最大。这是个最优化问题。按照（2）的需求使间距最大的时候，解需要遵循约束（1）。

求解这个问题的人类版本，就是拿一个尺子，尝试不同的直线来分隔所有点，直到你得到了使间距最大的那条。

人们发现，存在求解这个最大化的数学方式，但是它超出了我们的范围。为了进一步解释它，这里是个视频讲义，使用拉格朗日优化展示了它的工作原理。

你最后求解的超平面的定义，有关它相对于特定 `x_i` 的位置，它们就叫做支持向量，并且它们通常是最接近超平面的点。

(2) 如果你不能干净地分隔数据，会发生什么？

处理这个问题有两个方式。

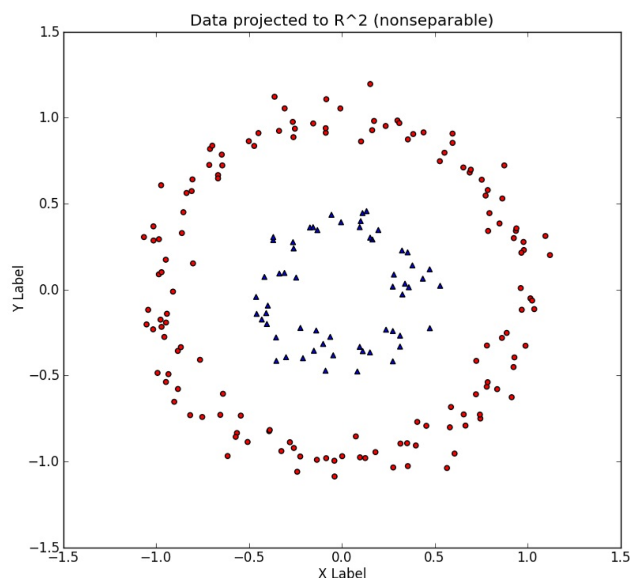
2.1 软化“分隔”的定义

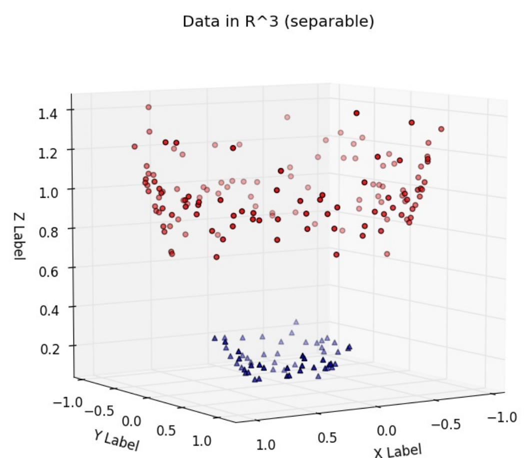
我们允许一些错误，也就是我们允许红色区域里面有一些蓝色点，或者蓝色区域里有一些红色点。我们向损失函数中。为错误分类的样本添加成本 `C` 来实现。基本上我们说，错误分类是可以接受的，只是会产生一些成本。

2.2 将数据放到高维

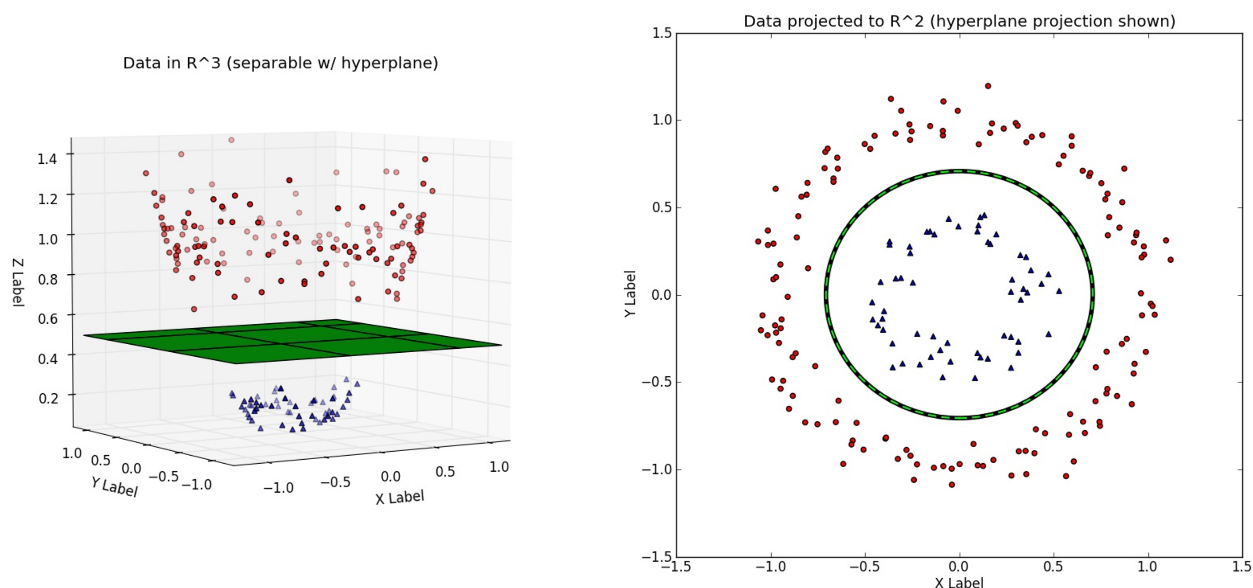
我们可以创建非线性的分类器，通过增加维数，也就是，包含 `x^2` ， `x^3` ，甚至是 `cos(x)` ，以及其它。突然，你就有了一个边界，当我们将其带回低维表示时，它看起来有些弯曲。

本质上，这就类似红的和蓝色的弹球都在地面上，它们不能用一条直线分隔。但是如果你让所有红色的弹球离开地面，像右图这样，你就能画一个平面来分隔它们。之后你让他们落回地面，就知道了蓝色和红色的边界。





> 二维空间 `R^2` 中的非线性可分的数据集，以及映射到高维的相同数据集，第三个维度是 `x^2+y^2` （来源：http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html）



> 决策边界展示为绿色，左边是三维空间，右边是二维空间。与上一张来源相同。

总之，SVM 用于二元分类。它们尝试寻找一个平面，干净地分隔两个类。如果这不可能，我们可以软化“分隔”的定义，或者我们把数据放到高维，以便我们可以干净地分隔数据。

好的！

这一节中我们涉及了：

- 监督学习的分类任务
- 两种基础的分类方法：对数几率回归（LR）和支持向量机（SVM）
- 常见概念：sigmoid 函数，对数几率（对率），以及假阳性（误报）和假阴性（漏报）

在“2.3：监督学习 III”中，我们会深入非参数化监督学习，其中算法背后的概念都非常直观，并且对于特定类型的问题，表现都很优秀，但是模型可能难以解释。

练习材料和扩展阅读

2.2a 对数几率回归

Data School 拥有一个对数几率回归的非常棒的深入指南。我们也继续向你推荐《An Introduction to Statistical Learning》。对数几率回归请见第四章，支持向量机请见第九章。

为了解释对数几率回归，我们推荐你处理这个问题集。你需要注册站点来完成它。很不幸，这就是人生。

2.2b 深入 SVM

为了深入 SVM 背后的数学，在 MIT 6.034：人工智能课程中观看 Patrick Winston 教授的讲义，并查看这个教程来完成 Python 实现。

2.3 监督学习 III

> 原文：Machine Learning for Humans, Part 2.3: Supervised Learning III
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
> 协议：CC BY-NC-SA 4.0
>
> 非参数化模型：KNN、决策树和随机森林。包含交叉验证、超参数调优和集成模型。

非参数学习器

> 事情变得有点...扭曲了。

我们目前为止涉及的方法，线性回归，对率回归和 SVM ，它们的模型形式是预定义的。与之相反，非参数学习器事先没有特定的模型结构。在训练模型之前，我们不会推测我们尝试习得的函数 `f` 的形式，就像之前的线性回归那样。反之，模型结构纯粹由数据定义。

这些模型对于训练数据的形状更加灵活，但是有时会有可解释性的代价。不久你就会更理解它。让我们继续吧。

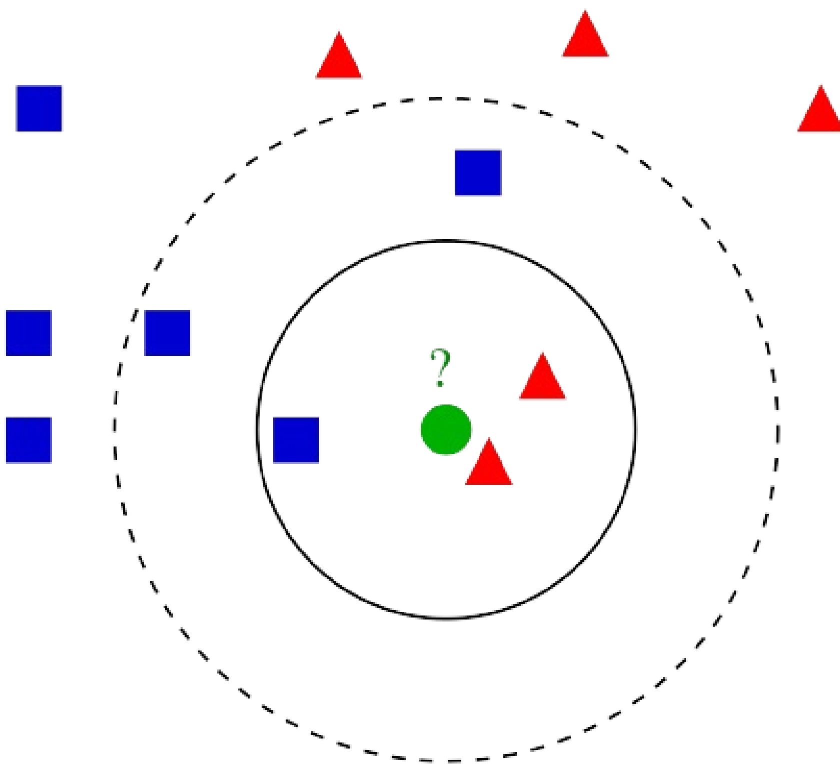
K 最近邻（KNN）

> 你是你的最亲密的 K 个朋友的均值。

KNN 看起来图样图森破，不像是机器学习算法。它的思路是，通过寻找 K 个最近的数据点的标签，来标记测试数据点 `x` 。

看一看下面的图像。让我们假设，你想知道，迷之绿色圆圈是红色三角还是蓝色方块。你怎么做呢？

你可以尝试提出一个迷之方程，它查看绿色圆圈在坐标平面的哪里，并作出相应的预测。或者，你可以仅仅查看三个最近的邻居，并猜测绿色圆圈可能是个红色三角。你也可以进一步扩展圆圈，并查看五个最近邻，并这样作出预测（五个最近邻里面，有三个蓝色方块，所以我们猜测， `k=5` 时迷之绿色圆圈是蓝色方块。



> KNN 的演示，其中`k=1, 3, 5`。为了划分上面的迷之绿色圆圈（`x`），查看它的单个最近邻，是个“红色三角”。所以我们猜测`ŷ`为“红色三角”。`k=3`时，查看三个最近邻：这里的众数仍然是“红色三角”，所以`ŷ`为“红色三角”。`k=5`时，我们选取五个最近邻的众数，要注意`ŷ`变为了“蓝色方块”。图片来自维基百科。

就是这样。这就是 KNN。你查看了 K 个最近的数据点，如果变量是连续的（例如房价），取它们的均值；如果变量是离散的（例如猫或者狗），取它们的众数。

如果你打算猜测未知房价，你可以选取一些地理上邻近的房子，然后取平均，你就会得到一些很棒的猜测。这可能甚至优于参数化回归模型，一些经济学家构建了它们来估计卧室/浴室、邻近的学校、公共交通的距离，以及其它的数量的参数。

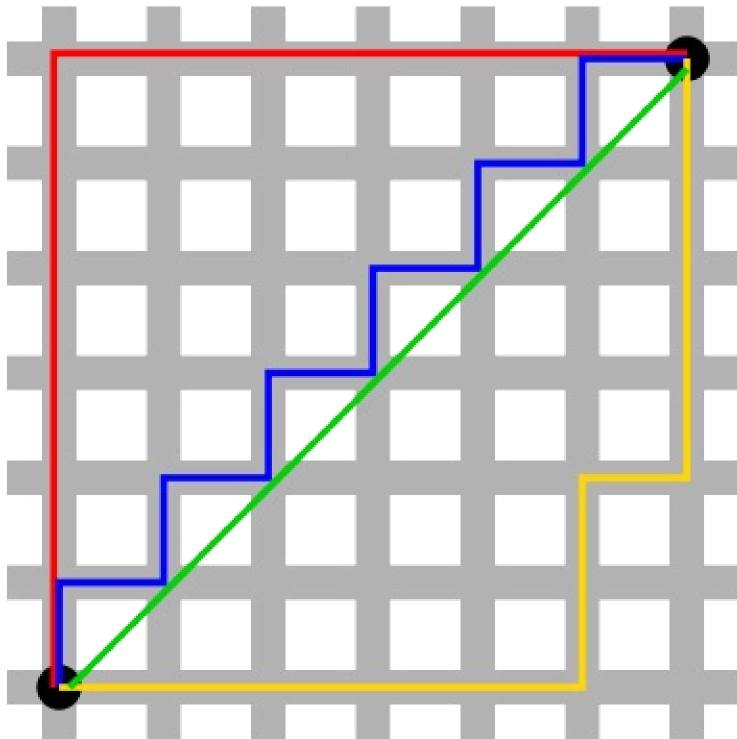
> 如何使用 KNN 来预测房价：
>
> 1) 储存训练集。`X`是特征，例如邮政编码、邻居、卧室数量、面积、公共交通的距离，以及其它。`Y`是对应的售价。
>
> 2) 将你的训练集排序，按照与测试集中的房子的相似性，基于`X`中的特征。我们下面会定义“相似性”。
>
> 3) 计算 K 个最邻近的房子的均值。这就是你对售价（也就是`ŷ`）的猜测。

KNN 不需要预定义的参数化函数`f(X)`，它用于将`Y`与`X`相关联。这使得它更适合关系过于复杂，不能用简单的线性模型表示的情况。

距离度量：定义和计算“邻近性”

在寻找“最近邻”的时候，你如何计算问题中的数据点的距离呢？你如何在数学上判断，示例中的哪个蓝色方块和红色三角更接近绿色圆圈？尤其是，如果你无法画出一幅漂亮的二维图像，用眼睛观测它？

最直接的度量是欧氏（几何）距离（“像乌鸦飞过”的一条直线）。另一个是曼哈顿（街区）距离，就像在城市块中行走。你可以想象，在涉及到 Uber 司机的费用计算的模型中，曼哈顿距离更加实用。



> 绿色直线为欧氏距离。蓝色直线为曼哈顿距离。来源：维基百科

还记得用于寻找直角三角形斜边长度的毕达哥拉斯（勾股）定理嘛？

$a^2 + b^2 = c^2.$

> `c` 为斜边（上面的绿色直线），`a` 和 `b` 是两个直角边（上面的红色直线）。

通过计算 `a` 和 `b` 长度的平方和的平方根，我们就解出了 `c`，求出了斜边长度。这里 `a` 和 `b` 是三角形的直角（正交）边（也就是，它们互为 90 度角，在空间中垂直）。

$c = \sqrt{a^2 + b^2}.$

给定两个正交方向的向量的情况下，求解斜边长度的思路，可以推广到多维。这就是 N 维空间的点 `p` 和 `q` 的欧氏距离 `d(p,q)` 的推导方式：

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

> 欧氏距离的公式，由勾股定理推出。

使用这个公式，你可以计算所有训练数据点，到你尝试标注的数据点的邻近度，并选取 K 个最近邻的均值或众数，来做出你的预测。

通常你不需要手动计算任何距离，用搜索引擎简单搜索一下，你就能在 NumPy 或者 SciPy 找到预构建的函数，会为你做这个事情，例如， `euclidean_dist = numpy.linalg.norm(p-q)` 。但是看到八年级的集合概念如何有助于构建当今的 ML 模型，这很有趣。

选取 `k` ：使用交叉验证调优超参数

为了决定我们使用哪个 `k` ，你可以测试不同的 KNN 模型，使用交叉验证以及 `k` 的不同值。

- 将你的训练集分成两部分，在一部分上训练模型，将保留的部分用作测试集。
- 通过将模型的预测（ `ŷ` ），与测试数据的真实值（ `y` ）相比，看看你的模型表现如何。
- 在所有迭代中，通常选取误差最小的模型。

Iteration 1	Train	Train	Train	Train	Test
Iteration 2	Train	Train	Train	Test	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Test	Train	Train	Train
Iteration 5	Test	Train	Train	Train	Train

> 交叉验证的演示。分块和迭代的数量可以修改。

K 的较高值防止过拟合

K 的较高值能防止过拟合，但是如果 K 太高的话，你的模型会有很大偏差，并且不灵活。选取一个极端的示例：如果 `k=N`（数据点的总数），模型就失效了，将所有测试数据分类为训练数据的均值或者众数。

如果动物数据集中的单个最常见的动物是苏格兰折耳猫，`k=N`（训练观测值数量）的 KNN 会将实际上的每个其它动物预测成它。在 Vishal 看来，这个很棒，但 Samer 不同意。



> 完全没有来由的苏格兰折耳猫 `.gif`。我们可以休息一下。

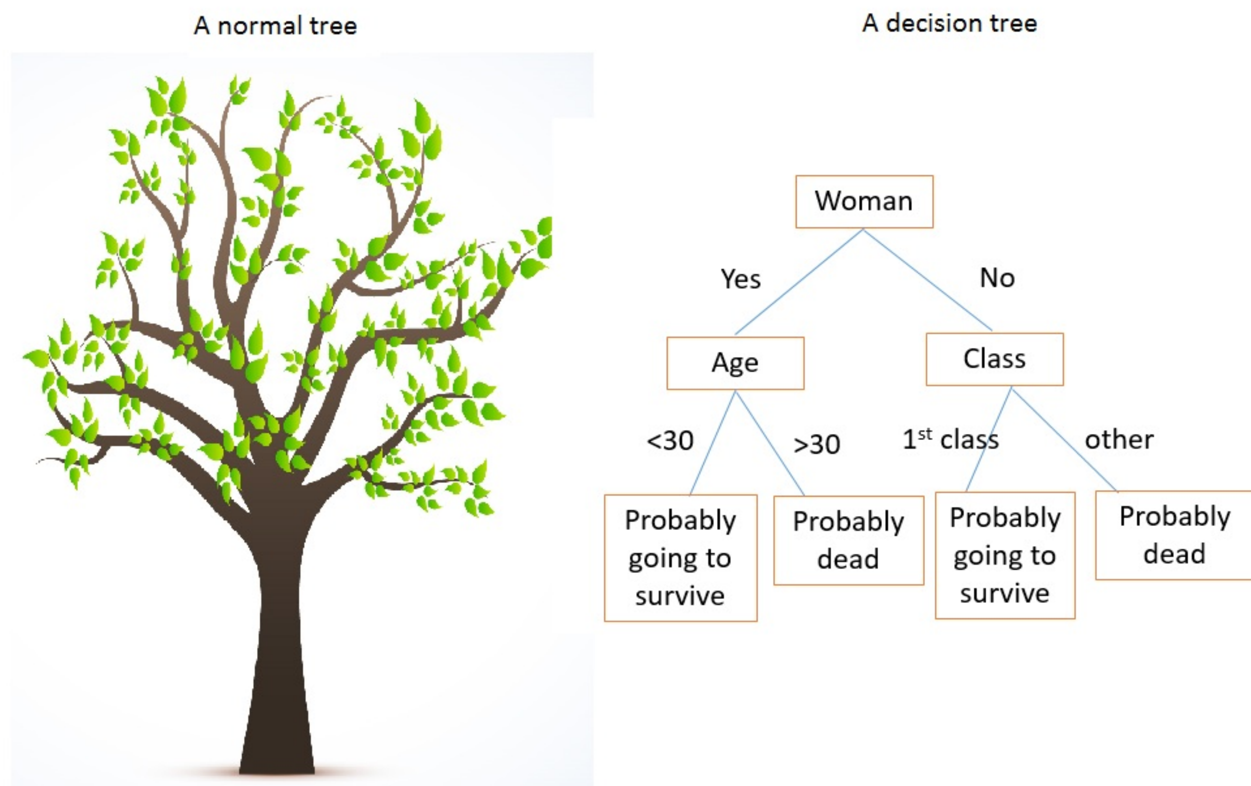
真实世界中使用 **KNN** 的地方

一些你可以使用 KNN 的地方：

- 分类：诈骗检测。模型可以使用新的训练样本马上更新，因为你仅仅是存储新的数据点，这允许你快速适应新的诈骗方法。
- 回归：预测房价。在房价预测中，字面上的“最近邻”实际上很好暗示了价格上的相似。KNN 在物理相似性很重要的领域很实用。
- 填充缺失的训练数据。如果你的 `.csv` 中的一列有大量缺失值，你可以通过选取均值或者众数填充数据。KNN 可能会给你每个缺失值的更加准确的猜测。

决策树和随机森林

> 制作一颗好的决策树就像玩“20个问题”的游戏。



> 右边的决策树描述了泰坦尼克号的生还者规律。

决策树根节点的第一次分割，就像是 20 个问题中的第一个问题：你打算尽可能干净地分隔数据，所以这个分割的信息增益应该最大。

如果你的朋友说，“我正在想一个名词，问我 20 个是或不是的问题来猜猜它”，并且你的第一个问题是“它是土豆嘛？”，你就太蠢了。因为如果它们说“不是”，你没有获得任何信息。除非你知道你的朋友整天都在想土豆，或者刚刚在想它，那么你就猜对了。

反之，类似“这是一个物体嘛？”的问题可能更有意义。

这有点类似医院分类病人，或者做出不同的诊断的方式。它们先问一些问题，并检查一些重要的指标来判断你是否马上就要挂了，还是不是。当你进门的时候，它们不会一开始就做切片检查，或者检查你是否得了胰腺癌。

有几种方式来量化信息增益，以便你能根本上求解训练集的每个可能的分割，以及每个分割的信息增益。你可以用这个方式，尽可能高效地预测每个标签或值。

现在，让我们查看一个特定的数据集，并讨论我们如何选择分割。

泰坦尼克数据集

Kaggle 的泰坦尼克数据集大量用于机器学习入门。当泰坦尼克沉没时，2224 个乘客和乘员中有 1502 个死亡。虽然包含一些运气成分，女人、孩子和头等舱更有可能生还。如果你回去看看上面的决策树，你会看到，它某些程度上反映了性别、年龄和舱位的变化。

选择决策树中的分割

熵是集合中的无序的总数，由基尼系数和交叉熵度量。如果值相当混杂，熵就很大；如果你能干净地分割数据，就没有熵。对于父节点的每个分割，你需要让子节点尽可能纯粹，也就是熵最小。例如，在泰坦尼克中，性别是生还者的重要决定因素，所以将这个特征用于第一次分割十分有意义，并且它是拥有最大信息增益的特征。

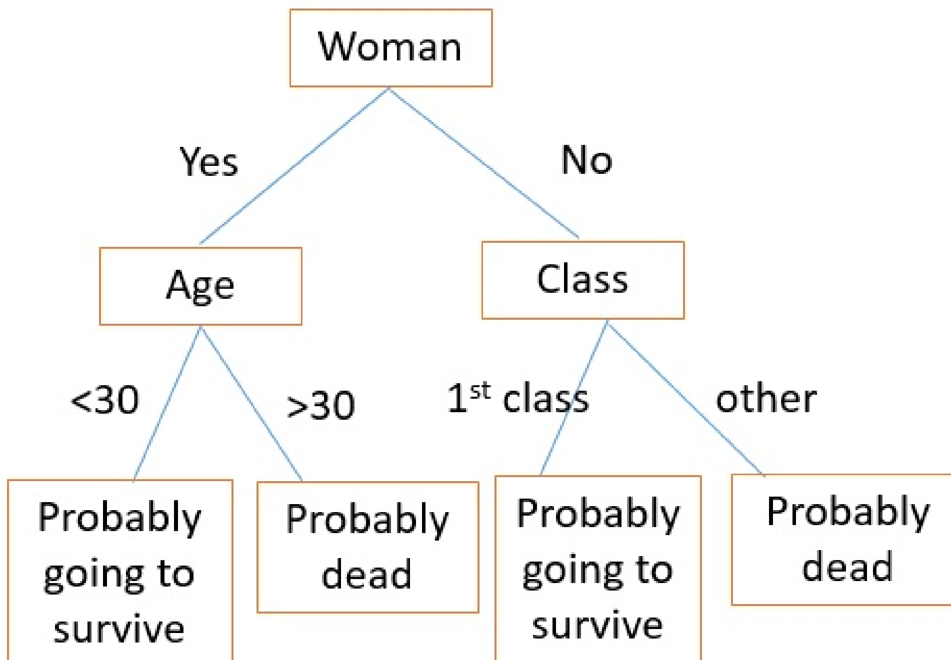
让我们看一看泰坦尼克的变量：

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

> 来源：Kaggle

我们通过选取这些变量之一，以及根据它分割数据集，来构建决策树。



第一次分割将我们的数据集分为男性和女性。之后女性分支又按照年龄分割（使熵最小的分割）。与之类似，男性分支按照舱位分割。通过对新的乘客遍历这棵树，你可以使用它来猜测它们是否挂了。

泰坦尼克的示例解决了分类问题（生存或者死亡）。如果我们将决策树用于回归，例如预测房价，我们可以分割决定房价的最重要的特征。面积：大于或小于 xxx？卧室或者浴室数量：大于或小于 xxx？

之后，在测试期间，你需要用特定的房子遍历所有分割，并取叶子节点的所有房价的均值（最底下的节点），这些房子就会成为你的售价预测。

决策树是高效的，因为它们易于解读，即使对凌乱的数据也很强大，并且一旦训练完成，部署的开销很小。决策树也擅长于处理混合数据（数值或类别）。

也就是说，决策树的训练开销很大，过拟合的风险很大，并且容易找到局部最优，因为它们在分割之后就不能回溯了。为了解决这些缺陷，我们转向了一种方式，它演示了将多个决策树整合为一个模型的力量。

随机森林：决策树的集成

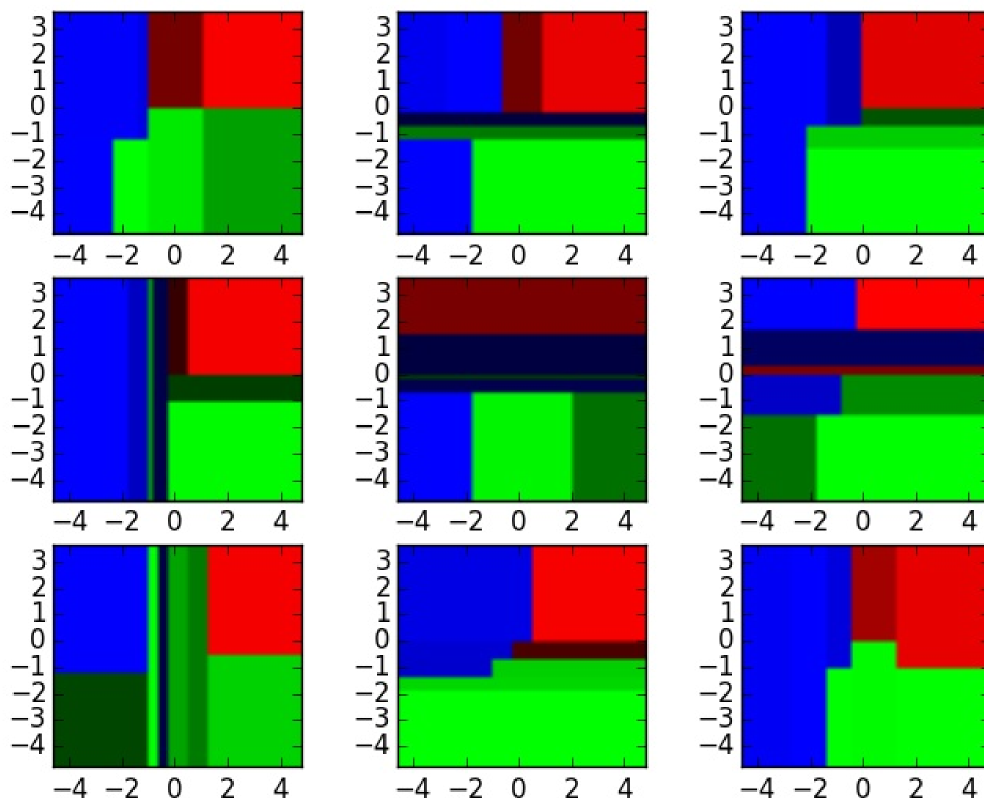
由多个模型整合的模型叫做集成模型，这通常是一个制胜策略。

单个决策树可能做出很多错误判断，因为它有很多非黑即白的判断。随机森林是个元估计其，它继承了多个决策树，并拥有一些可观的改进：

- 每个节点上分割的特征数量限制为总体的一定百分比（这是个可以调整的超参数，详见 scikit-learn 的文档）。这确保了继承模型不会过度依赖单个特征，并且公平地利用所有潜在的预测性特征。
- 每棵树从原始数据集抽取随机样本，来生成自己的分割，这添加了进一步的随机元素来防止过拟合。

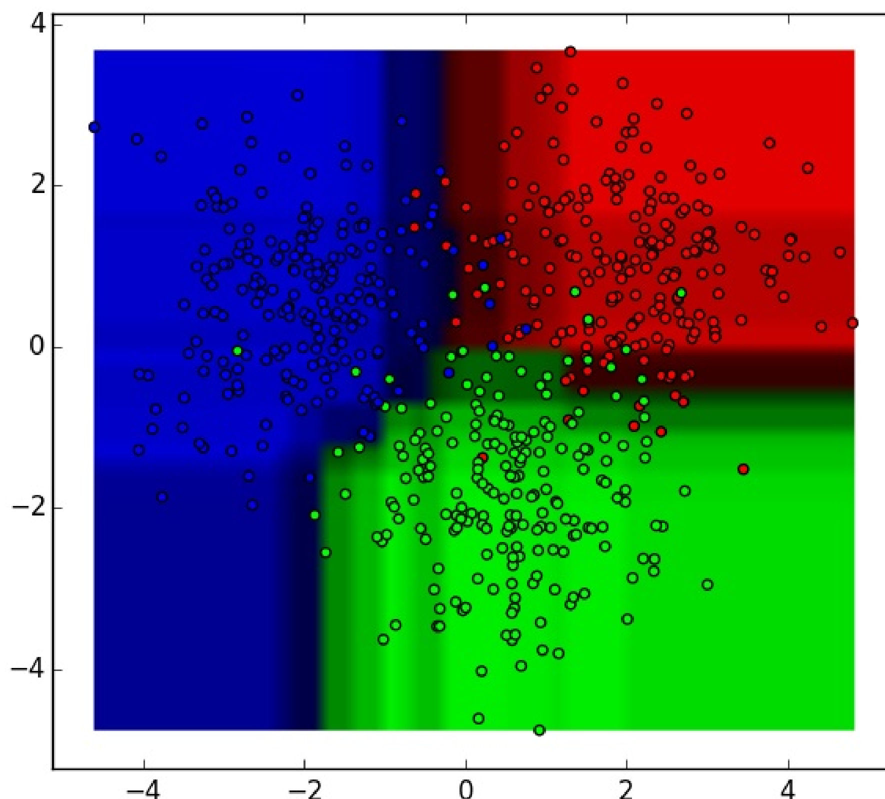
这些改进也防止了树的高度相关。如果没有上面的第一条和第二条，每棵树都是相同的，因为递归的二元分割时决定性的。

为了演示，看看下面的九个决策树分类器：



> 来源：http://xenon.stanford.edu/~jianzh/ml/

这些决策树分类器可以集成到随机森林中，它整合了这些输入。将横轴和纵轴看做特征 `x1` 和 `x2` 。对于每个特征的特定值，决策树输出“蓝色”、“绿色”和“红色”的分类。



这些结果通过众数（分类）或者均值（回归）整合为单个集成模型，它优于单个决策树的输出。

随机森林是建模过程的一个非常不错的起始点，因为它们对于不整洁的数据拥有高容忍度的强大表现。并且，对于在众多特征中找到最重要的特征非常实用。

也有很多机智的继承模型，它组合了决策树并产生非常棒的表现。请查看 XGBoost（eXtreme Gradient Boosting）的示例。

之后，我们就完成了监督学习的学习

非常不错。这一节中我们涉及了：

- 两个非参数监督学习算法：KNN 和决策树
- 距离和信息增益的度量
- 随机森林，它是集成模型的示例
- 交叉验证和超参数调优

我希望，你现在有了一些可靠的直觉，对于在给定训练集的情况下，我们如何习得 `f`，以及使用它和测试数据做出预测。

在“第三部分：无监督学习”中，我们讨论当我们拥有不带标签的训练集时，如何解决问题。

练习材料和扩展阅读

2.3a 实现 KNN

尝试这个攻略，在 Python 中从零开始实现 KNN。你可能也打算看一看 scikit-learn 的文档，来体验预构建的实现的工作方式。

2.3b 决策树

尝试《An Introduction to Statistical Learning》中的第八章的决策树实验。你可以使用泰坦尼克训练集来玩玩，并且查看这个教程，它涵盖了与上面相同的概念和代码。这里是随机森林的 scikit-learn 实现，可以在数据集上开箱即用。

三、无监督学习

> 原文： Machine Learning for Humans, Part 3: Unsupervised Learning
>
> 作者：Vishal Maini
>
> 译者：机器之心
>
> 聚类和降维：K-Means 聚类，层次聚类，主成分分析（PCA），奇异值分解（SVD）。

我们可以怎样发现一个数据集的底层结构？我们可以怎样最有用地对其进行归纳和分组？我们可以怎样以一种压缩格式有效地表征数据？这都是无监督学习的目标，之所以称之为「无监督」，是因为这是从无标签的数据开始学习的。

我们将在这里探索的两种无监督学习任务是：1）将数据按相似度聚类（clustering）成不同的分组；2）降维（reducing dimensionality），以便在保留数据结构和有用性的同时对数据进行压缩。

> 无监督学习方法可能有用的案例：
>
> - 一家广告平台需要根据相似的人口学特征和购买习惯将美国人口分成不同的小组，以便广告客户可以通过有关联的广告接触到他们的目标客户。
> - Airbnb 需要将自己的房屋清单分组成不同的社区，以便用户能更轻松地查阅这些清单。
> - 一个数据科学团队需要降低一个大型数据集的维度的数量，以便简化建模和降低文件大小。

和监督学习不同，要找到评价无监督学习算法优劣的指标可并不轻松。「表现水平」往往是主观的，而且因领域不同而各不相同。

聚类

聚类的一个有趣的真实应用案例是营销数据提供商 Acxiom 的人生阶段聚类系统 Personicx。这项服务将美国家庭分成了 70 个不同的聚类，它们分属于 21 个人生阶段分组，可以被广告主用于投放定向 Facebook 广告、陈列式广告和直邮广告等。

1Y STARTING OUT	8X LARGE HOUSEHOLDS	15M TOP WEALTH
Cluster 39 Setting Goals	Cluster 11 Schools & Shopping	Cluster 2 Established Elite
Cluster 45 Offices & Entertainment	Cluster 12 On the Go	Cluster 3 Corporate Connected
Cluster 57 Collegiate Crowd	Cluster 19 Country Comfort	
Cluster 58 Outdoor Fervor	Cluster 27 Tenured Proprietors	16M LIVING WELL
Cluster 67 First Steps		Cluster 14 Career Centered
	9B COMFORTABLE INDEPENDENCE	Cluster 15 Country Ways
2Y TAKING HOLD	Cluster 29 City Mixers	Cluster 23 Good Neighbors
Cluster 18 Climbing the Ladder	Cluster 35 Working & Active	
Cluster 21 Children First	Cluster 56 Metro Active	17M BARGAIN HUNTERS
Cluster 24 Career Building		Cluster 43 Work & Causes
Cluster 30 Out & About	10B RURAL-METRO MIX	Cluster 44 Open Houses
	Cluster 47 Rural Parents	Cluster 55 Community Life
3Y SETTLING DOWN	Cluster 53 Metro Strivers	Cluster 63 Staying Home
Cluster 34 Outward Bound	Cluster 60 Rural & Mobile	Cluster 68 Staying Healthy

> Personix 人口学特征聚类的一部分

他们的白皮书表明他们使用了重心聚类（centroid clustering）和主成分分析，这两种技术在这一节都有覆盖。

你可以想象，如果广告主想（1）理解他们已有的客户群，（2）通过相关的人口学特征、兴趣和生活习惯向潜在新客户投放定向广告以便高效利用广告开支，那么这些聚类将对他们非常有用。

What's My Cluster ?



Cluster #24: Career Building

Career Building singles are young, but well compensated. While repaying their education loans they are beginning to save and invest. They favor trendy stores that cater to their age range, incomes and aspirations, such as Express, H&M and Sephora. They enjoy new technology, and read magazines on mobile devices. They visit The Apple Store. They use the Internet extensively for entertainment news, music, podcasts and services. Sports are important, too, either as a fan or a participant. They listen to football, watch MMA and have fun skiing and playing volleyball.

> 实际上，你只需要在 Acxiom 的「我属于哪个聚类？」工具中回答几个简单问题，你就能知道你个人属于哪个聚类。

让我们了解几种聚类方法，看看这样的任务是如何完成的。

K 均值聚类

> 「重心之赛有 k 个魔戒，在那之上，是希望的力量。」

聚类的目标是为数据点分组，使得不同聚类中的数据点是不相似的，同一聚类中的数据点则是类似的。

使用 K 均值聚类，我们希望将我们的数据点聚类为 K 组。K 更大时，创造的分组就更小，就有更多粒度；K 更小时，则分组就更大，粒度更少。

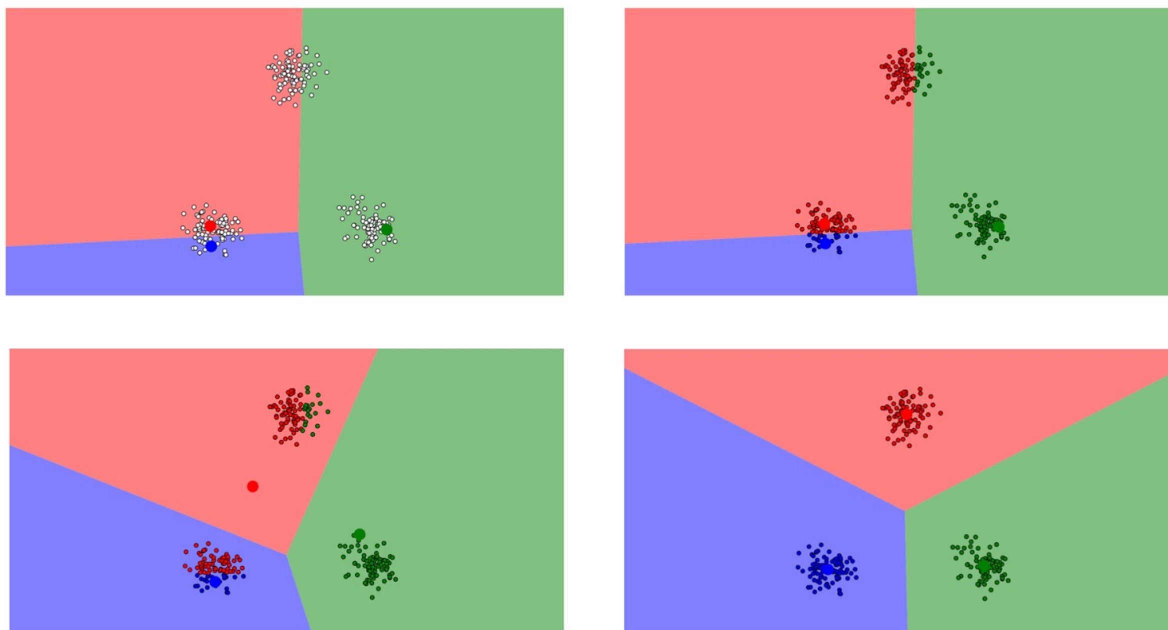
该算法的输出是一组「标签」，这些标签将每个数据点都分配到了 K 组中的一组。在 K 均值聚类中，这些组的定义方式是为每个组创造一个重心（centroid）。这些重心就像是聚类的心脏，它们可以「捕获」离自己最近的点并将其加入到自己的聚类中。

你可以把这些重心看作是派对上成为关注焦点的人，他们就像是有磁性一样。如果只有一个这样的人，每个人都会围绕在他周围；如果有很多这样的人，就会形成很多更小一点的活动中心。

> K 均值聚类的步骤如下：
>
> 1. 定义 K 个重心。一开始这些重心是随机的（也有一些更加有效的用于初始化重心的算法）
>
> 2. 寻找最近的重心并且更新聚类分配。将每个数据点都分配给这 K 个聚类中的一个。每个数据点都被分配给离它们最近的重心的聚类。这里的「接近程度」的度量是一个超参数——通常是欧几里得距离（Euclidean distance）。
>
> 3. 将重心移动到它们的聚类的中心。每个聚类的重心的新位置是通过计算该聚类中所有数据点的平均位置得到的。

重复第 2 和 3 步，直到每次迭代时重心的位置不再显著变化（即直到该算法收敛）。

这就是 K 均值聚类工作方式的精简版！该算法的可视化演示可在这里查看，你可以像读漫画一样理解。平面上的每个数据点都根据离自己最近的重心加了颜色。你可以看到这些重心（更大一点的蓝点、红点和绿点）一开始是随机的，然后很快进行了调整，得到了它们各自的聚类。



K 均值聚类的另一个真实应用是分类手写数字。假设我们有用像素亮度的长向量表示的数字的图像。假设这些图像是黑白两色的，大小为 `64x64` 像素。每个像素代表一个维度。那么这些图像就生活在一个有 `64x64=4096` 个维度的世界里。在这个 4096 维的世界里，K 均值聚类让我们可以按接近程度对这些图像分组，并且假设这些靠得很近的图像都是同一个数字。这种算法可以在数字识别上得到相当好的结果。

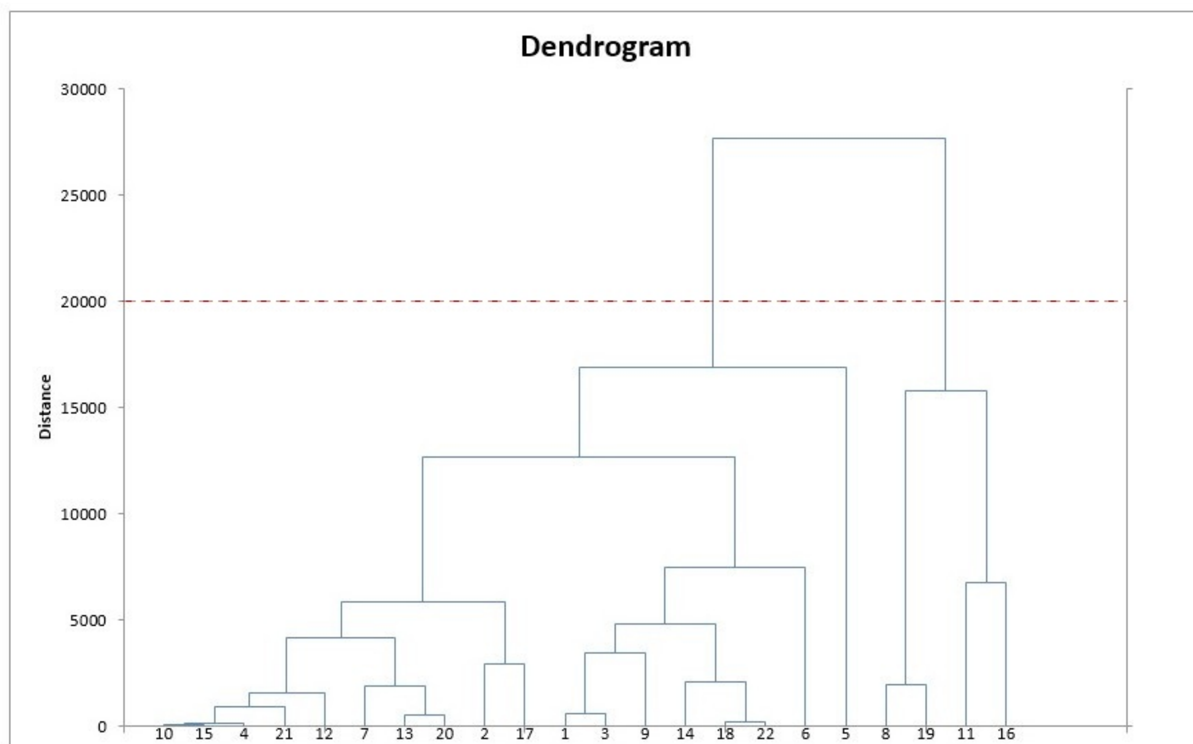
层次聚类

「让我们把 100 万个选项变成 7 个选项。或者 5 个。或者 20 个？呃，我们可以过会儿决定。」

层次聚类类似于常规的聚类，只是你的目标是构建一个聚类的层次。如果你最终的聚类数量不确定，那这种方法会非常有用。比如说，假设要给 Etsy 或亚马逊等网络市场上的项目分组。在主页上，你只需要少量大组方便导航，但随着你的分类越来越特定，你需要的粒度水平也越来越大，即区别更加明显的项聚类。

在算法的输出方面，除了聚类分配，你也需要构建一个很好的树结构，以帮助你了解这些聚类之间的层次结构。然后你可以从这个树中选择你希望得到的聚类数量。

> 层次聚类的步骤如下：
>
> 1. 首先从 `N` 个聚类开始，每个数据点一个聚类。
> 2. 将彼此靠得最近的两个聚类融合为一个。现在你有 `N-1` 个聚类。
> 3. 重新计算这些聚类之间的距离。有很多可以办到这件事的方法（参见这个教程了解更多细节）。其中一种方法（平均连接聚类，average-linkage clustering）是将两个聚类之间的距离看作是它们各自元素之间所有距离的平均。
> 4. 重复第 2 和 3 步，直到你得到包含 N 个数据点的一个聚类。你就会得到如下图所示的树（也被称为树状图））。
> 5. 选择一个聚类数量，然后在这个树状图中划一条水平线。比如说，如果你想要 `K=2` 个聚类，你应该在距离大约为 20000 的位置画一条水平线，你会得到一个包含数据点 8、9、11、16 的聚类和包含其它数据点的另一个聚类。一般而言，你得到的聚类的数量就是水平线与树状图中的竖直线的交叉点的数量。



> 来源：Solver.com。更多有关层次聚类的详细信息，可参阅这个视频。

降维

「对于那些该砍去的非精髓部分的态度，并不是每天增加吸收，而是每日尽量排减。」——李小龙

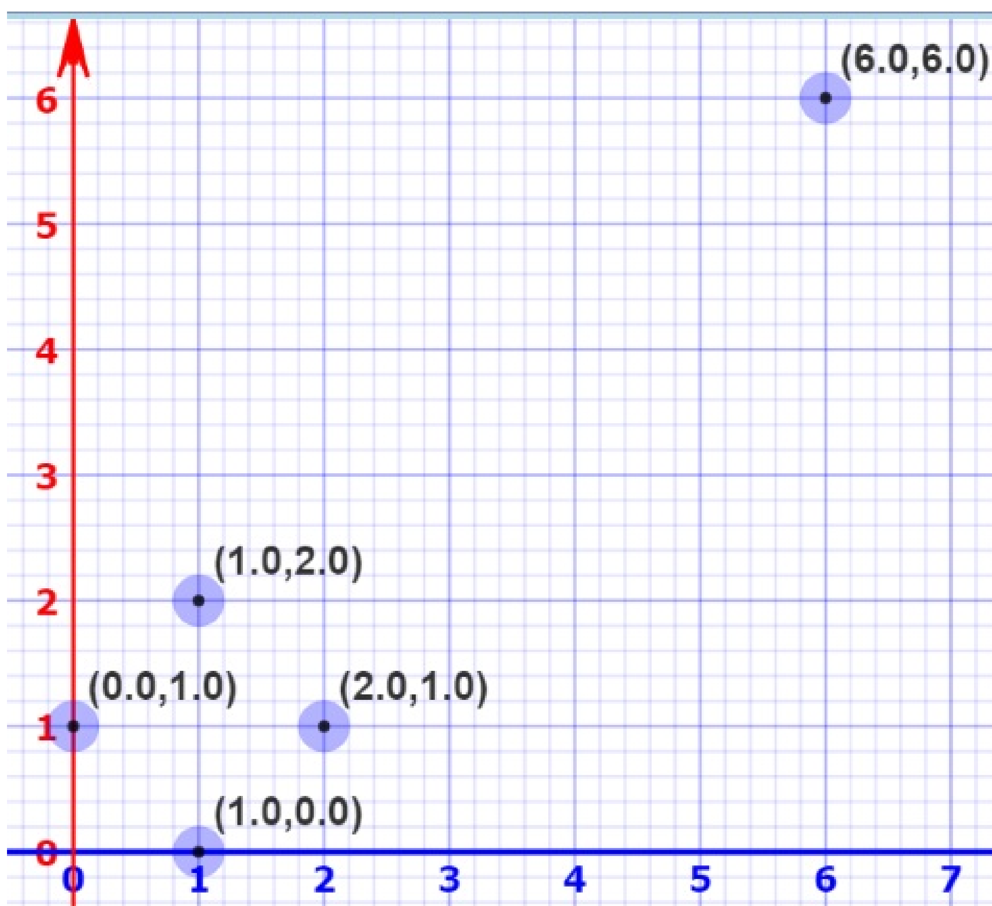
降维看上去很像压缩。这是为了在尽可能保存相关的结构的同时降低数据的复杂度。如果你有一张简单的 `128×128×3` 像素的图像（长×宽×RGB 值），那么数据就有 49152 维。如果你可以给这个图像空间降维，同时又不毁掉图像中太多有意义的内容，那么你就很好地执行了降维。

我们将了解两种实际中很常用的降维技术：主成分分析和奇异值分解。

主成分分析（PCA）

首先，了解一点线性代数知识——看看空间（space）和基（base）。

你应该知道由原点 `O(0,0)` 和基向量 `i(1,0)` 与 `j(0,1)` 定义的坐标平面。事实上，你也可以选择一个完全不同的基础，其中的数学仍然有效。比如说，你可以保持原点仍然为 O，但选择 `i'=(2,1)` 和 `j'=(1,2)` 作为基向量。如果你有耐心计算一下，你会发现在 `i', j'` 坐标系统中标记为 `(2,2)` 的点在 `i, j` 系统标记为 `(6, 6)` 。



> 使用 Mathisfun 的「交互式笛卡尔坐标」绘制

这意味着我们可以修改空间的基础。现在想象有更高维度的空间，比如有 5 万维。你可以为这个空间选择一个基础，然后根据这个基础仅选择 200 个最重要的向量。这些基向量被称为主成分，而且你可以选择其中一个子集构成一个新空间，它的维度比原来的空间少，但又保留了尽可能多的数据复杂度。

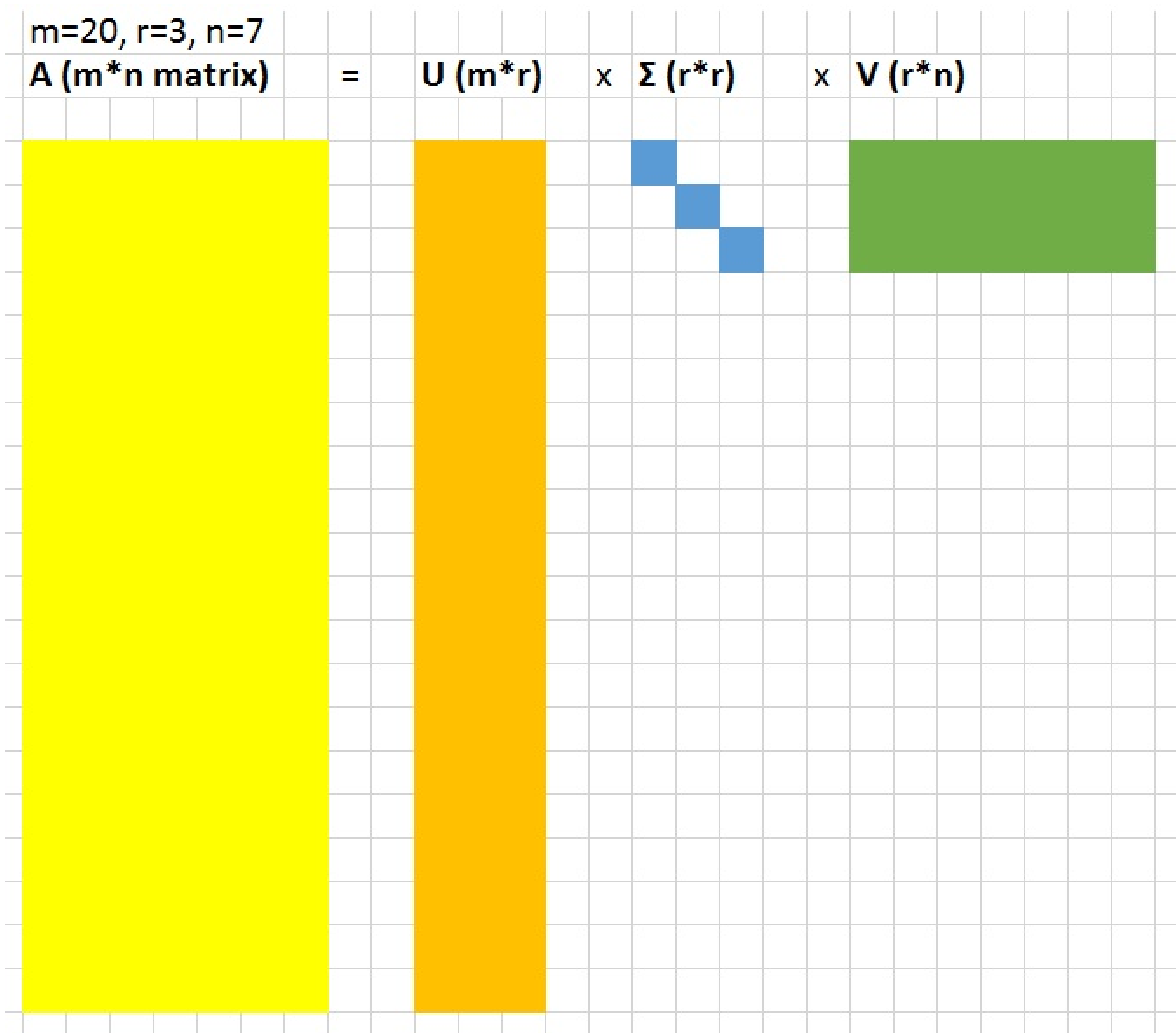
要选择出最重要的主成分，我们需要检查这些数据的方差，并按这个指标给它们排序。

理解 PCA 的另一个思路是 PCA 将我们数据中存在的空间重映射成了一个更加紧凑的空间。这种变换后的维度比原来的维度更小。

仅需使用重映射空间的前几个维度，我们就可以开始理解这个数据集的组织结构。这就是降维的目的：减少复杂度（即这里的维度），同时保留结构（方差）。这里有篇 Samer 写的论文，介绍了使用 PCA（以及扩散映射等技术）试图理解维基解密披露的电报。

奇异值分解（SVD）

假设我们将我们的数据表示成一个 `A=m×n` 的大型矩阵。SVD 让我们可以将这个大型矩阵分解成 3 个较小的矩阵的乘积；这 3 个矩阵分别是 `U=m x r` 、对角矩阵 `Σ=r x r` 、 `V=r x n` ，其中 `r` 是一个很小的值。

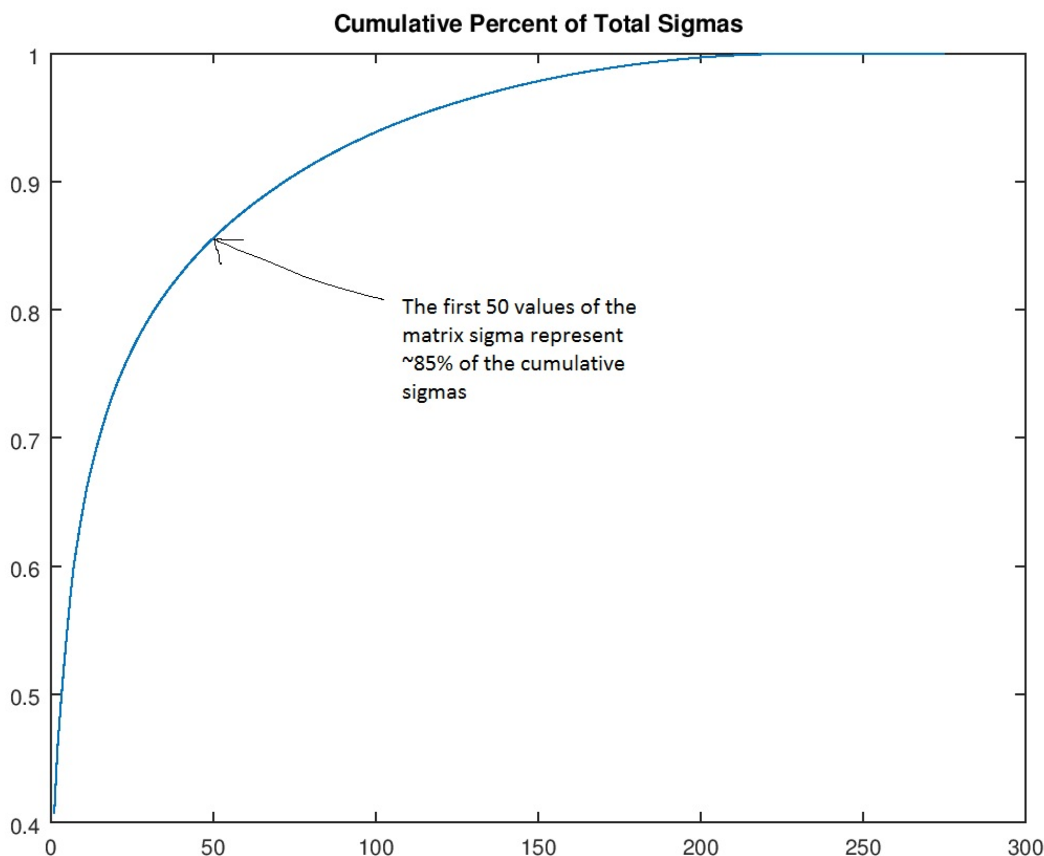


在这个 `r×r` 的对角矩阵 `Σ` 中的值被称为奇异值。这些值的奇妙之处是可以被用于压缩原来的矩阵，如果你丢弃奇异值中最小的 20% 以及矩阵 `U` 和 `V` 中相关的列，你就可以节省大量空间，同时仍然能很好地表征原来的矩阵。

为了更准确地了解其中的含义，我们来看看一张小狗的图片：



我们将使用 Andrew Gibiansky 写的关于 SVD 的文章中的代码。首先，我们发现如果我们根据大小排序这些奇异值（矩阵 Σ 的值），那么前 50 个奇异值将包含整个矩阵 Σ 的大小的 85%。



根据这个事实，我们可以丢弃后面的 250 个值（即将它们设为 0），仅保留这张小狗图像的「rank（秩）50」版本。这里，我们创建了秩为 200、100、50、30、20、10 和 3 的小狗照片。显然，照片变小了。但假设我们认为秩为 30 的小狗仍然很好，现在让我们看看我们实现了多少压缩。原先的图像矩阵有 `305*275 = 83,875` 个值，秩为 30 的图像则有 `305*30+30+30*275=17,430` 个值。值的数量差不多少了 5 倍，但质量却下降很少。上述计算的原因是当我们执行 `UΣ'V` 运算时，`U` 和 `V` 矩阵中的一部分因为乘 0 也被丢弃（其中 `Σ'` 是 `Σ` 的修改后版本，其中仅包含了前面的 30 个值）。

Full-Rank Dog



Rank 200 Dog



Rank 100 Dog



Rank 50 Dog



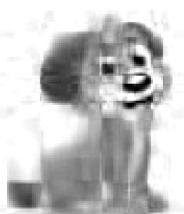
Rank 30 Dog



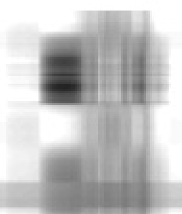
Rank 20 Dog



Rank 10 Dog



Rank 3 Dog



无监督学习常常被用于数据预处理。一般而言，这意味着以某种平均-保留的方式压缩数据，比如 PCA 或 SVD；之后，这些数据可被用于深度神经网络或其它监督式学习算法。

请继续！

既然你完成了这一章，你已经得到了一个糟糕的、永远不会再次提及的无监督学习的笑话。这就是：

> Person-in-joke-#1: Y would u ever need to use unsupervised tho?
>
> Person-in-joke-#2: Y? there’s no Y.

下面是第四章：神经网络和深度学习。

实践材料和扩展阅读

3a K-Means 聚类

玩一玩这个聚类的演示，来构建算法工作原理的直觉。之后，看一看这个手写数字的 K-Means 聚类的实现，以及相关教程。

3b SVD

对于 SVD 的不错的参考，没有东西比 Andrew Gibiansky 的文章更好了。

四、神经网络和深度学习

> 原文：Machine Learning for Humans, Part 4: Neural Networks & Deep Learning
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
>
>
> 深度神经网络的工作地点、原因和方式。从大脑中获取灵感。卷积神经网络（CNN）和循环神经网络（RNN）。真实世界中的应用。

使用深度学习，我们仍然是习得一个函数`f`，将输入`X`映射为输出`Y`，并使测试数据上的损失最小，就像我们之前那样。回忆一下，在 2.1 节监督学习中，我们的初始“问题陈述”：

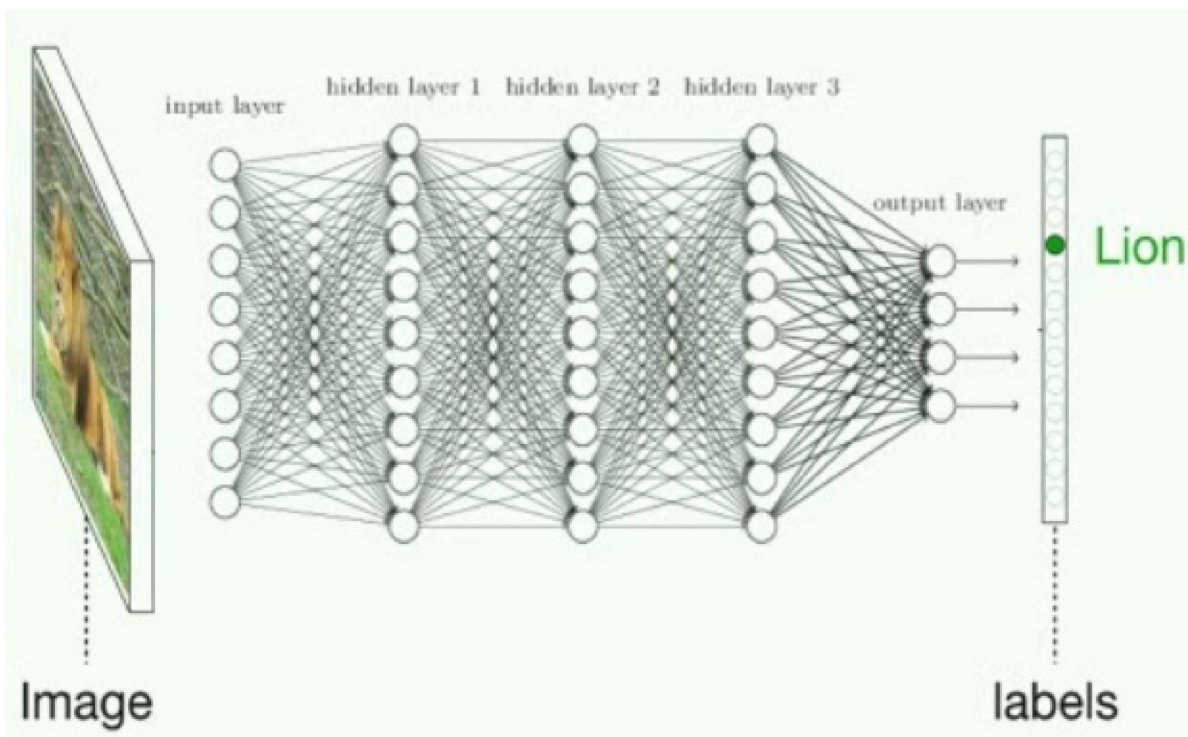
> `Y = f(X) + ϵ`
>
> 训练：机器从带标签的训练数据习得`f`
>
> 测试：机器从不带标签的测试数据预测`Y`

真实世界很乱，所以有时`f`很复杂。在自然语言问题中，较大的词汇数量意味着大量特征。视觉问题设计大量的像素相关的视觉信息。玩游戏需要基于复杂场景做决策，也带有许多可能的未知。当我们处理的数据不是很复杂时，我们目前涉及的学习机制做的很好。但是，它们如何推广到像这样的场景，还不清楚。

深度学习非常善于习得`f`，特别是在数据很复杂的场景中。实际上，人工神经网络也被成为通用函数近似器，因为它们能够学习任何函数，无论多么扭曲，都只需要一个隐藏层。

让我们看看图像分类的问题。我们选取一张图片作为输入，并输出一个分类（例如狗、猫、车）。

通过图解，深度神经网络的图像分类解决方式，类似于这样：



> 图片来自 Jeff Clune 在 YouTube 上的一小时深度学习概览

但是说真的，这是个巨大的数学方程，有数百万个项和大量参数。假设输入 `X` 是灰度图像，由 `w*h` 的像素亮度矩阵表示。输出 `Y` 是每个分类的概率的向量。也就是说，我们输出“每个分类是正确标签”的概率。如果这个神经网络能用的话，最高的概率就是正确的分类。然后，中间的层仅仅计算大量的矩阵相同，通过在每个隐藏层之后，使用非线性变换（激活函数），对激活值 `x` 的权重求和，来让神经网络习得非线性函数。

难以置信的是，你可以使用梯度下降，以 2.1 节中我们对线性回归所做的相同方式，使损失最小。所以使用大量样本和大量梯度下降，模型可以习得如何正确分类动物图片。这个，简单来说，就是“深度学习”。

深度学习做得好的地方，以及一些历史

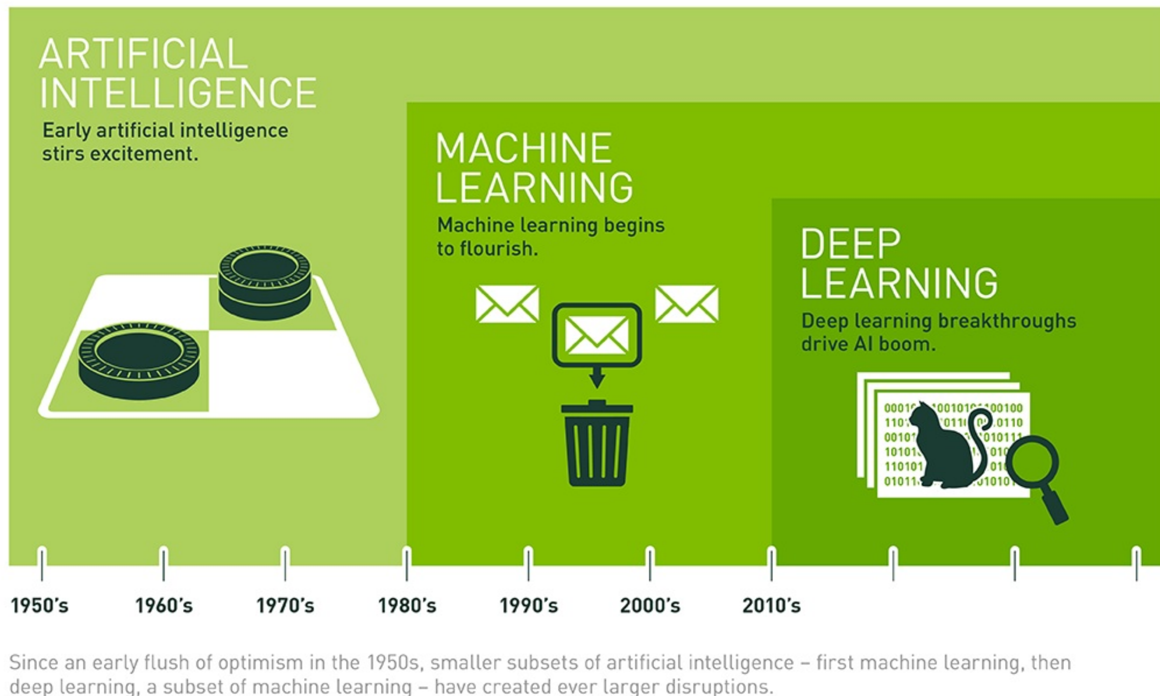
人工神经网络实际上有很长时间了。它们的应用可追溯到控制论（cybernetics，1940s~1960s），连接机制（connectionism，1980s~1990s），之后变成了流行的深度学习。大约在 2006 年，神经网络开始变得“更深”（Goodfellow 等，2016）。但是最近我们才开始触及它们的全部潜能。

就像 Andrej Karpathy（特斯拉的 AI 总监，它被认为是深度学习的 Shaman）描述的那样，有四个“将 AI 带回来”的不同因素：

- 计算（最明显的一个：摩尔定律，GPU，ASIC）
- 数据（以不错的形式，并不在互联网上某处，也就是 ImageNet）
- 算法（研究和理念，也就是 backprop, CNN, LSTM），以及
- 基础设施（你下面的软件，Linux, TCP/IP, Git, ROS, PR2, AWS, AMT, TensorFlow, 以及

其它）（Karpathy, 2016）。

在过去的十年当中，深度学习的全部潜能，最终被（1）和（2）的进步解锁，它反过来产生了（3）和（4）的突破。并且这个循环仍在继续，成倍的人们沿着这条路（想想你现在做的事情），不断参与到深度学习研究的前线中。



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

> 由 NVIDIA 演示，它是一家~~核弹~~ GPU 领导厂商。GPU 最开始为游戏构建，但是最终很好地顺应了深度学习所需的并行计算。

在这一章的剩余部分中，我们提供一些来自生物和统计的被寂静，来解释神经网络里面发生额了什么，之后讨论深度学习的一些神奇应用。最后，我们链接到一些资源，所以你可以自行应用深度学习，甚至穿着睡衣坐在沙发上，拿着笔记本，在特定种类的问题上快速实现超越人类级别的表现。

从大脑中（或者只是统计？）获取灵感：神经网络中发生了什么

神经元、特征学习和抽象层次

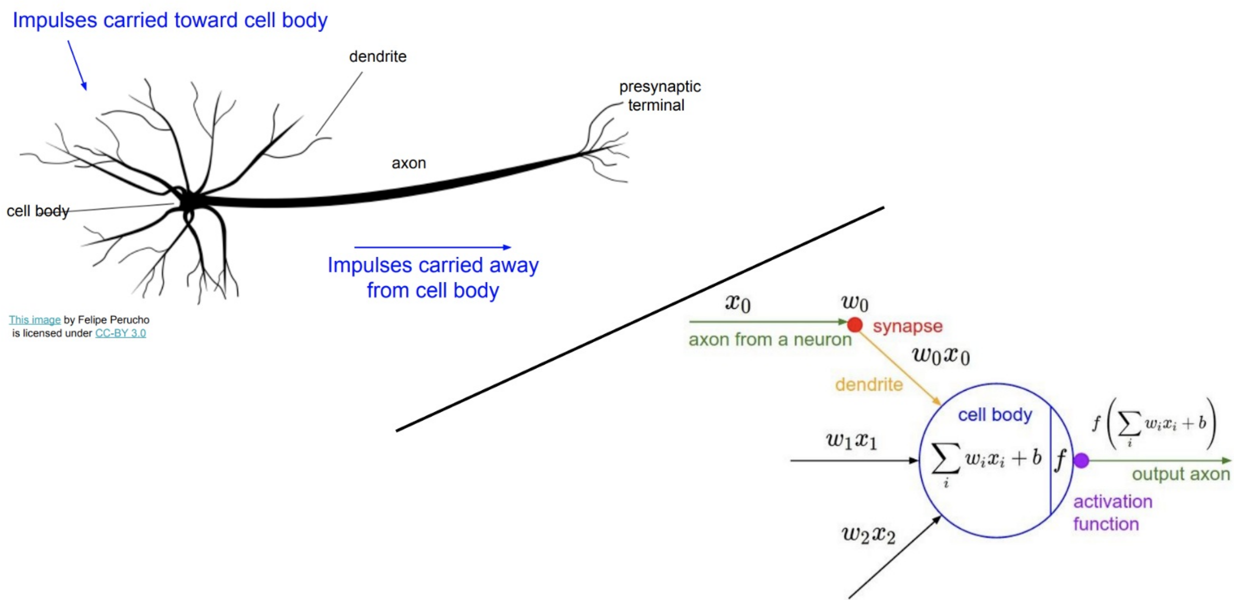
当你阅读这些词汇的时候，你并没有检查每个词的每个字，或者组成字的每个像素，来获得单词的含义。你从细节中将东西抽象、组装成高阶的概念：词、短语、句子、段落。

> Yuor abiilty to exaimne hgiher-lveel fteaures is waht aollws yuo to unedrtsand waht is hpapening in tihs snetecne wthiout too mcuh troulbe (or myabe yuo sned too mnay dnruk txets).

（你检测高阶特征的能力，让你理解句子中发生的东西，而不会太麻烦（或者可能你发送了大量的乱序文本）。

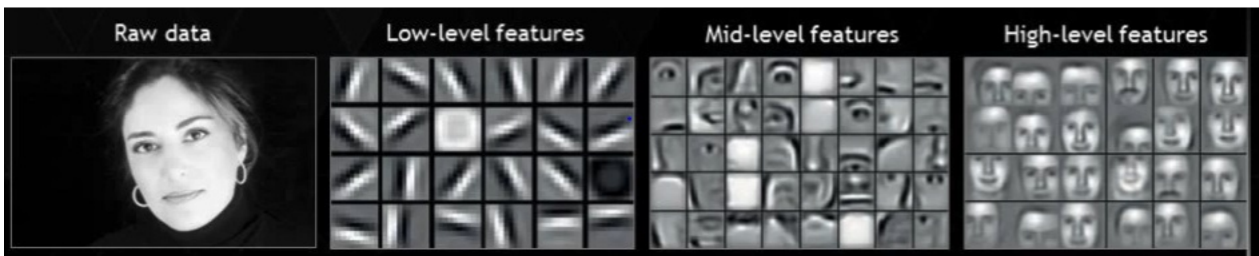
视觉上发生的事情相同，这并不仅仅在人类，也在动物的视觉系统中。

大脑由神经元组成，它们在足够“激活”之后，通过向其他神经元发送电信号来“触发”。这些神经元都是可塑的，根据有多少来自神经元的信号添加到当前神经元的激活水平上（大概来说，将神经元彼此连接的权重最后会被训练，使神经连接更加有用，就像线性回归中的参数可以被训练，来改进输入到输出的映射）。

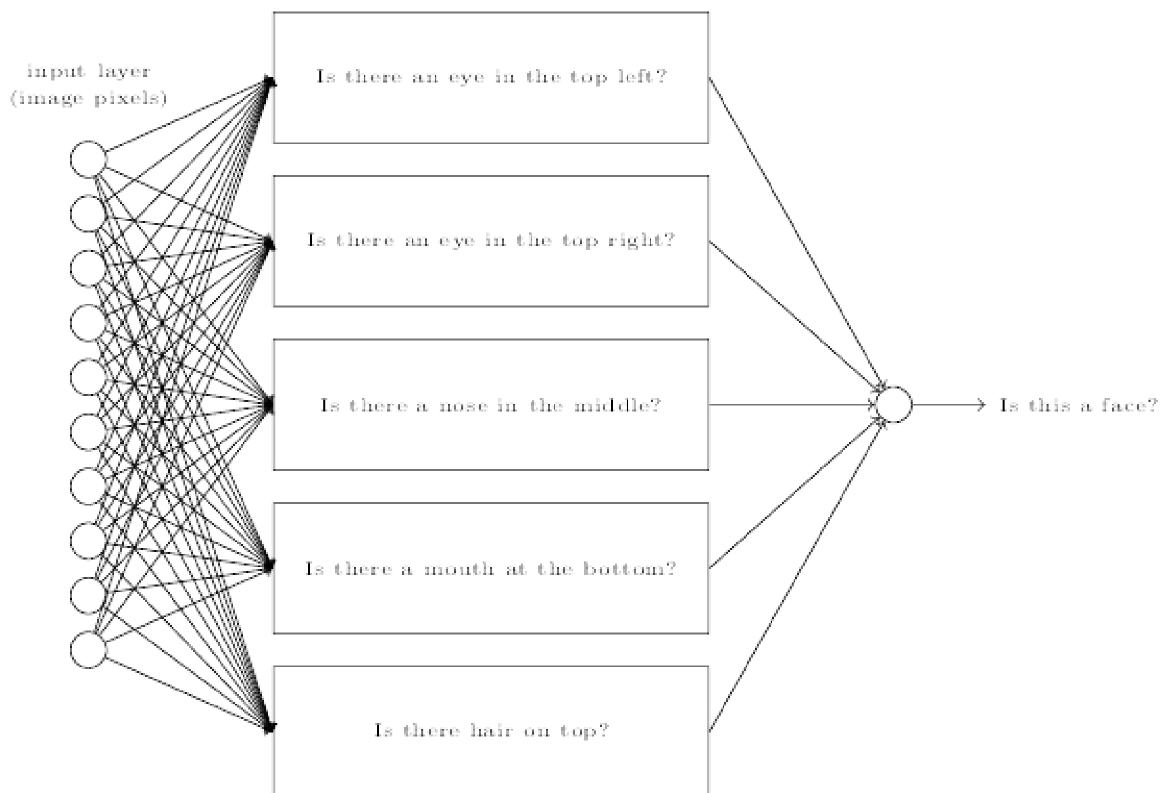


> 生物和人工神经网络的端到端的演示，来自斯坦福 CS231n。这个比喻不是很严谨，生物神经元可以做人工神经元不能做的事，反之亦然。但是理解生物中的灵感十分有用。更多细节请见生物和人工神经元的维基百科的描述。

我们的生物网络以层次方式排列，以便特定神经元最终检测我们周围的世界的，不是极其特定的特征，而是更加抽象的特征，也就是，更低阶的特征的规律或者分组。例如，人类视觉系统中的纺锤状脸部区域可特别用于人脸识别。



> 持续学习抽象特征的演示，来自 NVIDIA。



> 人工神经网络如何选取原始像素输入，开发中介“神经元”来检测高阶特征（也就是鼻子的存在），以及组合它们的输出来创建最终输出。来自《神经网络和深度学习》（Nielsen, 2017）。

生物神经网络表现出的层次结构最早于 1950s 发现，当研究员 David Hubel 和 Torsten Wiesel 研究猫的视觉皮层中的神经元的时候。在给猫大量刺激之后，它们不能观察到神经活动：暗点、亮点、挥手、甚至是杂志上的女性照片。但是在它们的失望中，它们从投影仪中移除了对角线处的照片，它们注意到了一些神经活动。结果是，位于某个特定角度的，对角线上的边能够导致特定的神经元被激活。

https://medium.com/media/a9eddb961e4bc9ca2d118b64525c2659?postId=cdad8aeae49b

> 图片来自 Knowing Neurons

这样就逐渐有意义了，因为自然环境通常是嘈杂和随机的（想想草原和荒野）。所以当野外的猫感知到一条“边”的时候，也就是一条不同于背景的线，这可能表明，一个物体或者生物进入了视野范围。当边缘的神经元的特定组合被激活时，这些活动会集体产生更加抽象的活动，以此类推，直到最终的抽象是个有用的概念，比如“鸟”或者“狼”。

深度神经网络背后的概念就是，使用人工神经网络的层次来模拟类似的结构。

为什么线性模型不能用

为了提炼斯坦福的优秀的深度学习课程，CS231n：卷积神经网络和视觉识别，想象我们打算训练一个神经网络，使用下列标签的正确的那个来分类图像：`["plane", "car", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck"]`。

一种方式可能是，使用训练样本为每个分类构造一个“模板”，或者平均图像，之后使用最近邻算法，在测试期间来度量每个未分类图像的像素值，到每个模板的距离。总的来说是这样。这个方法不涉及任何抽象层次。这是个线性模型，将每个图像类型的所有不同的方向组合为一个平均的影子。

例如，它可以选取所有的车，无论它们面向左边、右边、中间，也无论它们的颜色，只是将它们平均。模板最终看起来非常模糊。



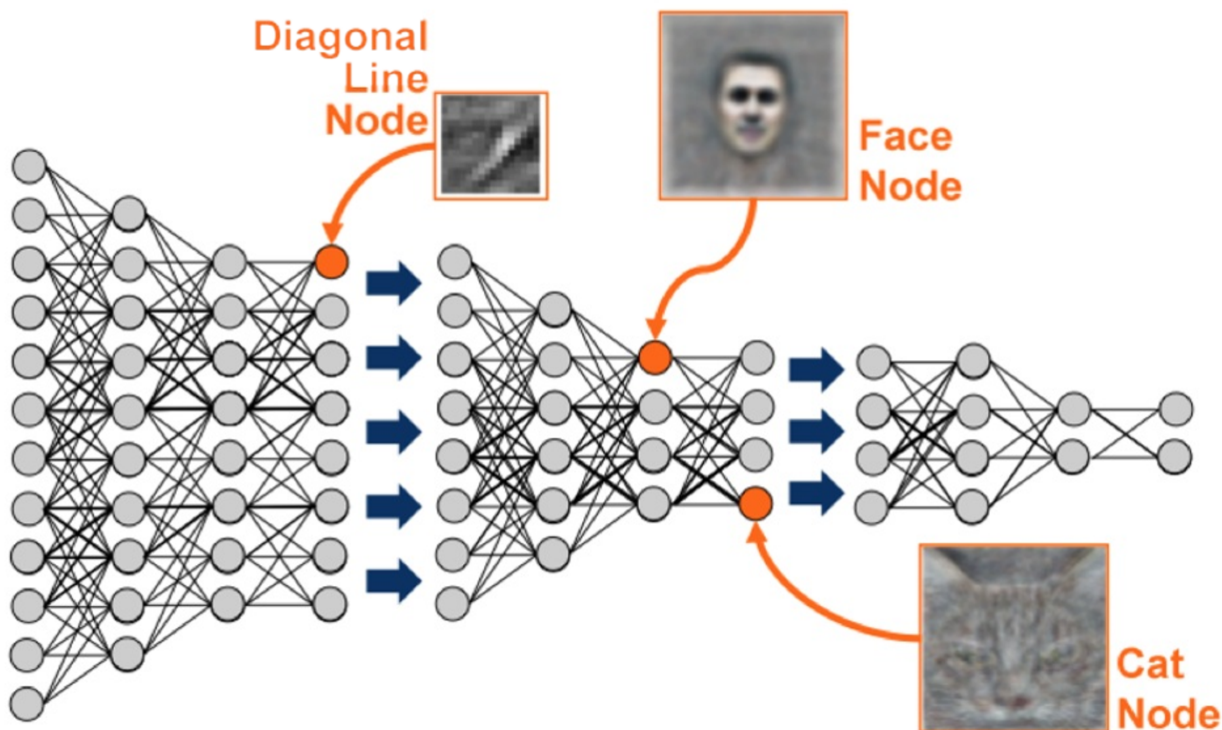
> 来自斯坦福 CS231n 第二篇讲义的样本。

要注意，上面的马的模板出现了两个头。这对我们并没什么帮助：我们想要能够分别检测头朝右边或者头朝左边的马，并且如果这些特征的任何一个被检测到了，我们就说我们看到了一匹马。深度神经网络提供给我们这种灵活性，我们会在下一节中看到。

深度神经网络使用抽象层来解决图像分类问题。

为了重复我们在这一节之前解释的东西：输入层接受图像的原始像素亮度。最终的层级会是类概率的输出向量（也就是图像是猫、车子、马，以及其他的概率）。

但是我们不习得一个简单的，和输入输出相关的线性模型。我们构造网络中间的隐藏层，它们会渐进学习抽象特征，这让我们不会丢失复杂数据中的所有细微差异。



> 来源：Analytics Vidhya

就像我们描述的动物大脑检测抽象特征，隐藏层中的人工神经元会学着检测抽象概念，无论哪个概念，只要它对于捕捉最多信息，以及使网络输出的准确度中的损失最小非常实用（这是个发生在网络中的，无监督学习的实例）。

这损失了模型的可解释性，因为当你添加更多隐藏层时，神经元开始表示越来越多的抽象和无法理解的特征。在这个层面上，你可能听说，深度学习也被称作“黑箱优化”，其中你基本上只是随机尝试一些东西，然后观察出现了什么，而无需真正理解里面发生了什么。

线性回归是可解释的，因为你决定了模型中包含哪个特征。深度神经网络难以解释，因为习得的特征在任何地方都不能用自然语言解释。它完全在机器的想象中。

一些值得注意的扩展和深层概念

- 深度学习软件包。你很少需要从零开始实现神经网络的所有部分，因为现有的库和工具使深度学习更加易于实现。有许多这类东西：TensorFlow, Caffe, Torch, Keras, 以及其它。
- 卷积神经网络（CNN）。CNN 特地为接受图像输入而设计，并且对于计算机视觉任务非常高效。它们也有助于深度增强/强化学习。CNN 的灵感特别来源于动物的视觉皮层的工作方式，并且它们是深度学习课程的热点。我们已经在文章中引用它了，就是斯坦福 CS231n。
- 循环神经网络（RNN）。RNN 拥有内建记忆的概念，并且非常适合语言问题。它们在强化学习中也很重要，因为它们让智能体能够跟踪东西在哪里，以及之前发生了什么，甚至在那些东西不能一次性看到的时候。Christopher Olah 编写了一个非常棒的，在语言问题的语境中的， RNN 和 LSTM 的参考文献。
- 深度增强/强化学习。这是深度学习研究中的最刺激的领域之一，处在近期研究的核心位置，例如 OpenAI 击败了 Dota2 职业玩家，以及 DeepMind 的 AlphaGo 在围棋竞技中胜过人类。我们在第五章会深度，但是本质上的目标是将这篇文章中的所有技术，应用于一个问题，教会智能体使回报最大。这可以用于与任何可以游戏化的环境，从真实的游戏，例如反恐精英或者吃豆人，到无人驾驶的汽车，或者股票交易，最终到真实生活和真实世界。

深度学习应用

深度学习正在重构世界的几乎每个领域。这里是深度学习可以做的无法置信的事情的示例：

- Fackbook 训练了一个神经网络，由短时记忆加持，来智能地回答《指环王》情节中的问题。

Facebook AI Research
@FBAIResearch

Home
About
Photos
Reviews
Videos
Posts
Community
Create a Page

Here is an example of what the system can do. After having been trained, it was fed the following short story containing key events in JRR Tolkien's Lord of the Rings:

Bilbo travelled to the cave.
Gollum dropped the ring there.
Bilbo took the ring.
Bilbo went back to the Shire.
Bilbo left the ring there.
Frodo got the ring.
Frodo journeyed to Mount-Doom.
Frodo dropped the ring there.
Sauron died.
Frodo went back to the Shire.
Bilbo travelled to the Grey-havens.
The End.

After seeing this text, the system was asked a few questions, to which it provided the following answers:

Q: Where is the ring?
A: Mount-Doom
Q: Where is Bilbo now?
A: Grey-havens
Q: Where is Frodo now?
A: Shire

> 来自 FAIR（Facebook AI 实验室）的研究，将加持了分离的短时记忆的深度神经网络，应用于回答 LOTR 情节的问题。这是史诗级的定义。

- 无人驾驶的汽车依赖深度学习，用于解决视觉任务，例如理解路面标志，检测道路和识别障碍。



> 来源：Business Insider

- 深度学习可以用于有趣的东西，例如艺术生成。一个叫做 Neural Style（神经风格）的工具可以模拟艺术家的风格，并且用它来重新组合另一幅图片，令人难以置信。



> 梵高的《星夜》的风格应用于斯坦福校园的图像上，通过 Justin Johnson 的 neural style 实现：https://github.com/jcjohnson/neural-style

其它值得注意的示例包括：

- 预测药物发现中的分子的生物活性。
- 用于照片和视频标注的人脸识别。
- 增强谷歌搜索的结果。
- 自然语言理解和生成，也就是谷歌翻译
- 火星探索者机器人“好奇号”，基于视觉检测，自动选取值得检查的土壤目标。

...以及很多，非常多，太多了。

现在开始去做吧！

对于神经网络如何实际建立，我们这里没有涉及太多细节。因为你通过自己实现，更易于理解细节。这里是一些非常棒的实践资源，用于起步。

- 使用谷歌的 Neural Network Playground 玩转神经网络的架构，来看看不同的配置如何影响网络的性能。
- 使用这篇谷歌的教程快速准备好并起步：TensorFlow 和深度学习。以超过 99% 的准确度分类手写数字，你不需要是 PhD，只需要在三个小时内，熟悉 TensorFlow 以及学习深度学习概念。
- 之后，浏览斯坦福 CS231n 的至少前几篇讲义，以及第一个作业，从零开始构建双层的神经网络，来实际强化这篇文章中涉及的概念。

更多资源

深度学习是个非常广阔的领域。因此，我们也编译了一些最佳资源，我们在这个话题中遇到过它们，以便你打算深入了解。

- Deeplearning.ai
    - 中文地址
- CS231n：卷积神经网络和视觉识别，斯坦福的深度学习课程。我看过的最佳课程，带有不错的讲义和演示性问题集。
    - 中文地址
- 神经网络和深度学习，易读，但很严谨。
    - 中文地址
- 深度学习圣经，基础，更加数学化。
    - 中文地址
- Fast.ai，不是很理论化，更加应用和黑箱。
- 查看 Greg Brockman 对“作为工程师，什么是修炼深度学习技能的最佳方式？”问题的回答，在 Quora 上。

下一章：到了玩游戏的时候了！

最后是第五章：强化/增强学习。

五、强化学习

> 原文：Machine Learning for Humans, Part 5: Reinforcement Learning
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
> 协议：CC BY-NC-SA 4.0
>
> 探索和利用。马尔科夫决策过程。Q 学习，策略学习和深度强化学习。
>
> 我刚刚吃了一些巧克力来完成最后这部分。

在监督学习中，训练数据带有来自神一般的“监督者”的答案。如果生活可以这样，该多好！

在强化学习（RL）中，没有这种答案，但是你的强化学习智能体仍然可以决定如何执行它的任务。在缺少现有训练数据的情况下，智能体从经验中学习。在它尝试任务的时候，它通过尝试和错误收集训练样本（这个动作非常好，或者非常差），目标是使长期奖励最大。

在这个“写给人类的机器学习”的最后一章中，我们会探索：

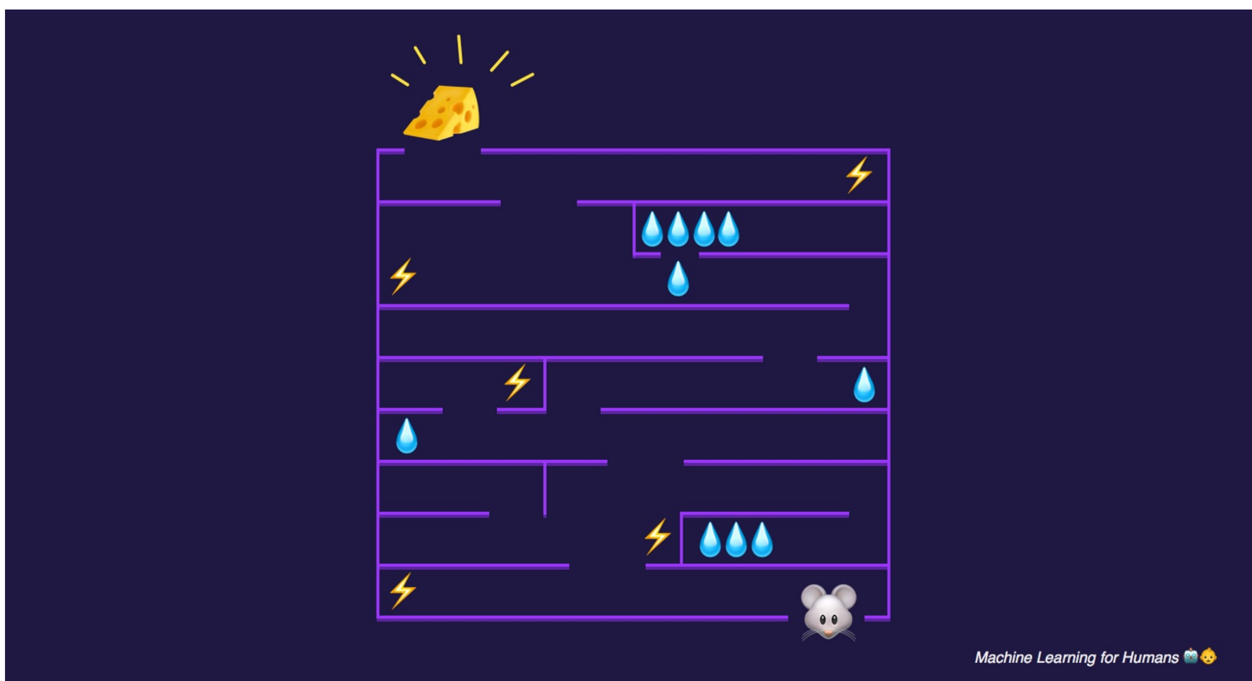
- 探索和利用的权衡
- 马尔科夫决策过程（MDP），用于 RL 任务的经典配置
- Q 学习，策略学习和深度强化学习
- 最后，价值学习的问题

最后，像往常一样，我们编译了一些最喜欢的资源，用于深入探索。

让我们在迷宫中放一个机器老鼠

思考强化学习的最简单的语境是一个游戏，它拥有明确的目标和积分系统。

假设我们正在玩一个游戏，其中我们的老鼠正在寻找迷宫的尽头处的奶酪的终极奖励（+ 1000 分），或者沿路的水的较少奖励（+ 10 分）。同时，机器老鼠打算避开带有电击的区域（⚡ - 100 分）。



> 奖励就是奶酪

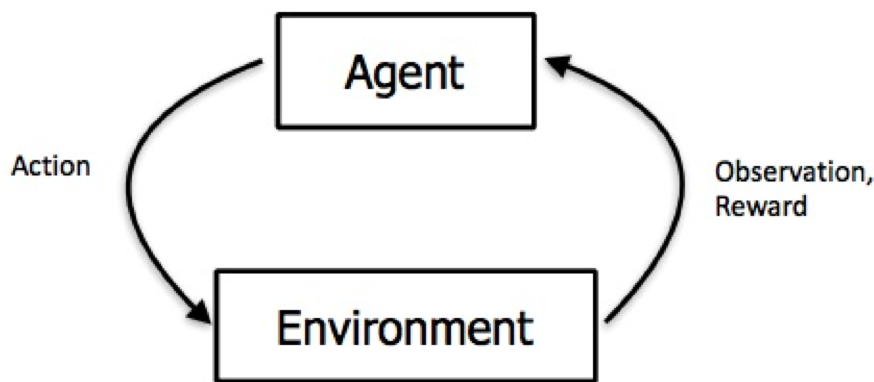
在一些探索之后，老鼠可能找到三个水资源的小型天堂，并且花费它的时间来利用它的发现，通过不断积累水资源的小型奖励，永远不深入迷宫来追求更大的奖励。

但是你可以看到，老鼠会错误迷宫深处的一片更好的绿洲，它就是尽头处的奶酪的终极奖励。

这就产生了探索和利用的权衡。老鼠的一种用于探索的简单策略是，在大多数情况下（可以是 80%），做出最佳的已知动作，但是偶尔探索新的，随机选取的方向，即使它可能远离已知奖励。

这个策略叫做 epsilon 贪婪策略，其中 epsilon 就是“给定全部已知知识的条件下，智能体做出随机选取的动作，而不是更可能最大化奖励的动作”的时间百分比（这里是 20%）。我们通常以大量探索起步（也就是较高的 epsilon 值）。一会儿之后，随着老鼠更加了解迷宫，以及哪个操作产生更大的长期奖励，它会将 epsilon 逐渐降到 10%，或者甚至更低，因为它习惯于利用已知。

重要的是要记住，奖励并不总是立即的：在机器老鼠的示例中，迷宫里可能有狭长的通道，你需要走过它，在你到达奶酪之前可能有好几个决策点。



> 智能体观测环境，做出动作来与环境互动，并接受正向或者负向的奖励。图片来自 UCB CS 294：深度强化学习，由 John Schulman 和 Pieter Abbeel 讲授

马尔科夫决策过程

老鼠迷宫之旅可以形式化为马尔科夫决策过程。这是一个过程，状态到状态的转移拥有特定的概率。我们会通过参考我们的机器老鼠的示例来解释。MDP 包含：

- 有限的状态集。我们的老鼠在迷宫中有可能的位置。
- 每个状态上的可用的动作集。这个就是走廊中的“前进，后退”，或者十字路口中的“前进，后退，左转，右转”。
- 状态之间的转换。例如，如果在十字路口左转，你就会到达一个新的位置。一系列概率可能链接到至少一个状态上（也就是，当你在口袋妖怪游戏中使用招式的时候，可能没打中，造成一些伤害，或者造成足够的伤害来击倒对手）。
- 和每个转换有关的奖励。在机器老鼠的例子中，多数奖励都是 0，但是如果你到达了一个位置，那里有水或者奶酪，就是正的，如果有电击就是负的。
- 0 和 1 之间的折现系数 `γ`。它量化了立即奖励和未来奖励的重要性的差异。例如，如果 `γ` 是 0.9，并且 3 步之后奖励为 5，那么奖励的当前值就是 `0.9 ** 3 * 5`。
- 无记忆。一旦知道了当前状态，老鼠的历史迷宫踪迹可以擦除，因为马尔科夫的当前状态包含来自历史的所有拥有信息。换句话说，“了解现在的情况下，未来不取决于历史”。

既然我们知道了 MDP 是什么，我们可以形式化老鼠的目标。我们尝试使长期奖励之和最大。

$$\sum_{t=0}^{t=\infty} \gamma^t r(x(t), a(t))$$

让我们逐项观察这个和式。首先，我们所有时间中对步骤 `t` 求和。让我们在这里将 `γ` 设为 1 并忘掉它。`r(x,a)` 是奖励函数。对于状态 `x` 和动作 `a`（也就是在十字路口左转），它会带给你奖励，和状态 `s` 上的动作 `a` 相关。回到我们的方程，我们尝试使未来奖励的和最大，通过在每个状态做出最佳动作。

既然我们建立了我们的强化学习问题，并形式化了目标，让我们探索几种可能的解决方案。

Q 学习：学习动作-分值函数

Q 学习是一种技巧，它基于动作-分值函数求解了要做出哪个动作，这个函数确定了在特定状态下做出特定行为的分值。

我们拥有一个函数 `Q`，它接受一个状态和一个动作作为输入，并返回这个动作（以及所有后续动作）在这个状态上的预期奖励。在我们探索环境之前，`Q` 提供相同（任意）的固定值。但是之后，随着我们探索了更多环境，`Q` 向我们提供动作 `a` 在状态 `s` 上的分值的，不断优化的近似。我们在这个过程中更新我们的函数 `Q`。

这个方程来自维基百科的 Q 学习页面，很好解释了它。他展示了，我们如何更新 Q 的值，基于我们从环境中得到的奖励：

$$Q(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_{a} Q(s_{t+1},a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t,a_t)}_{\text{old value}} \right)$$

让我们忽略折现系数 `γ`，再次将其设为 1。首先要记住，Q 应该为所有奖励之和，来自所选动作 Q 和所有后续的最优动作。

现在让我们从左到右浏览方程。让我们在状态 `st` 上做出动作，我们更新我们的 `Q(st,at)` 的值，通过向其添加一项。这一项包含：

- 学习率 `alpha`：这表示在更新我们的值时，我们有多激进。当 `alpha` 接近 0 时，我们更新得不是很激进。当 `alpha` 接近 1 时，我们简单将原值替换为新的值。
- 奖励 `reward` 就是我们通过在状态 `st` 做出动作 `at` 得到的奖励。所以我们将这个奖励添加到原有的估计中。
- 我们也添加了估计的未来奖励，它就是 `xt+1` 上的所有可用动作的，最大的可实现的奖励 `Q`。
- 最后，我们减掉原有值 `Q`，来确保我们仅仅增加或减少估计值的差（当然要乘上 `alpha`）。

既然对于每个状态-动作的偶对，我们拥有了值的估计，我们可以选取要做出哪个动作，根据我们的动作-选取策略（我们每次不一定选择导致最大预期奖励的动作，也就是使用 epsilon 贪婪探索策略，我们以一定百分比做出随机的动作）。

在机器老鼠的例子中，我们可以使用 Q 学习来找到迷宫中每个位置的分值，以及每个位置上动作“前进，后退，左转，右转”的分值。之后我们可以使用我们的动作-选取策略，来选择老鼠在每一步实际上做什么。

策略学习：状态到动作的映射

在 Q 学习方式种，我们习得了一个分值函数，它估计了每个状态-动作偶对的分值。

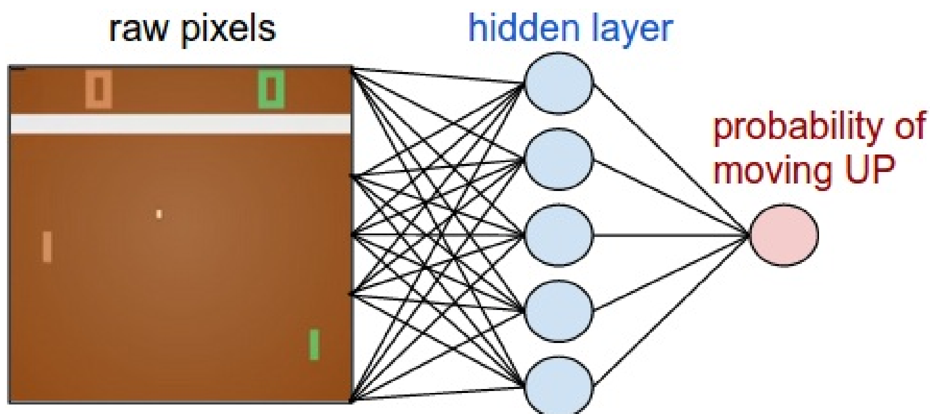
策略学习是个更直接的替代，其中我们习得一个策略函数 `π`，它是每个状态到最佳对应动作的直接映射。将其看做一个行为策略：“当我观测到状态 `s` 时，最好执行动作 `a`。”例如，一个自动驾驶的策略可能包括：“如果我看到黄灯，并且我离十字路口超过 100 英尺，我应该停下来。否则，继续向前移动。”

$$a = \pi(s)$$

> 策略是状态到动作的映射

所以我们习得了一个函数，它会使预期奖励最大。我们知道，什么最擅长习得复杂的函数呢？深度神经网络！

Andrej Karpathy 的 Pong from Pixels 提供了一个杰出的示例，关于习得一个用于 Atari 游戏 Pong 的策略，它接受来自游戏的原始像素作为输入（状态），并输出向上或向下移动拍子的概率（动作）。



> 在策略梯度网络中，智能体习得最优策略，通过基于来自环境的奖励信号，使用梯度下降来调整它的权重。图片来自 http://karpathy.github.io/2016/05/31/rl/

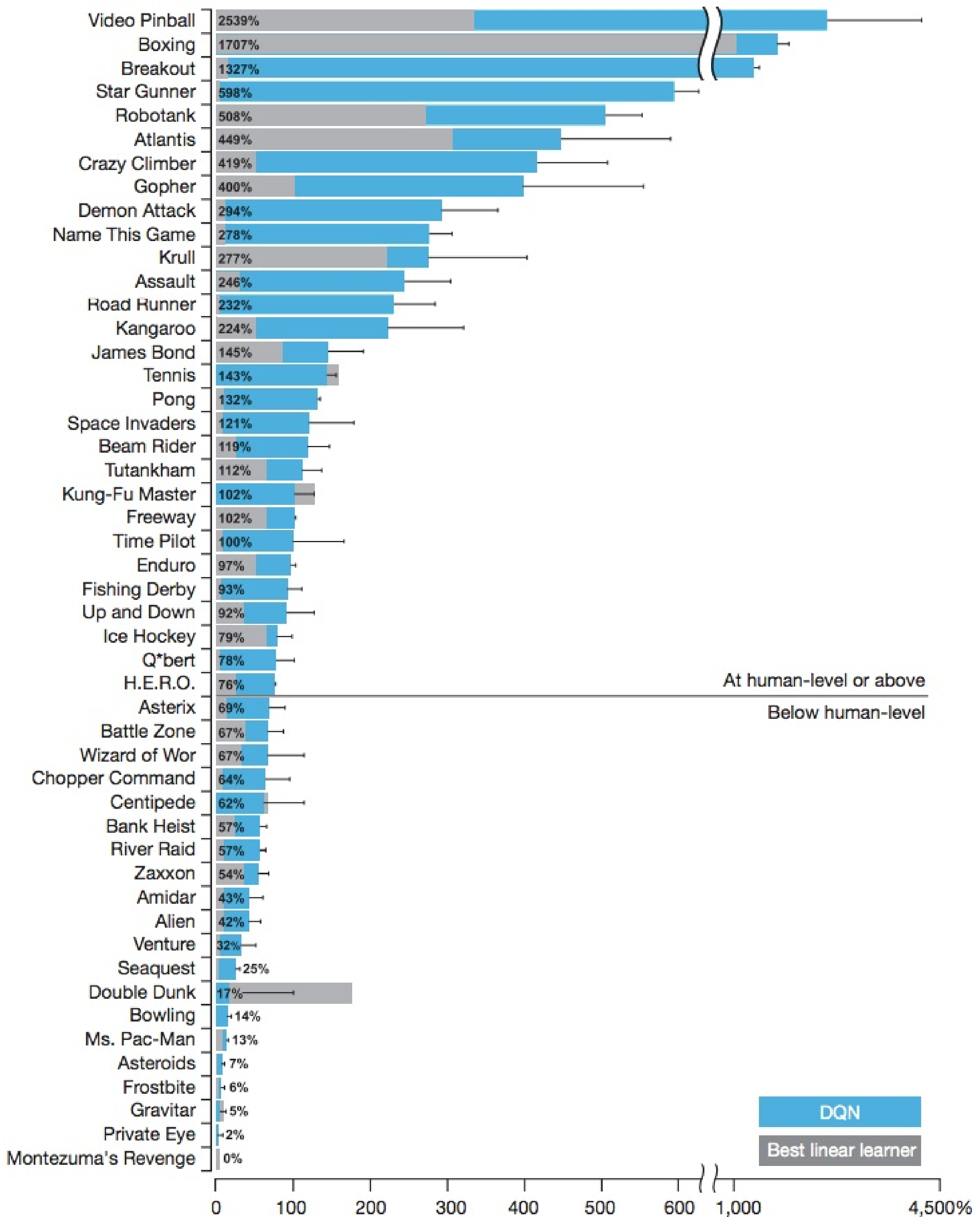
如果你打算亲自试一试深度 RL，查看 Andrej 的文章。你会在 130 行代码内实现一个二层的策略网络，并且会学到如何切入 OpenAI 的训练场，它允许你实现并运行你的第一个强化学习算法，在大量游戏上测试它，并且查看它的表现与其它记录相比怎么样。

DQN，A3C，和深度 RL 中的进展

在 2015 年，DeepMind 使用了一个叫做深度 Q 网络（DQN）的方法，使用深度神经网络近似 Q 函数，以便在许多 Atari 游戏中击败人类：

> 我们展示了深度 Q 网络的智能体，仅接收像素和游戏得分作为输入，能够超越所有以前的算法的表现，并在一组 49 个游戏中，达到专业人类游戏测试人员的相当水平，使用相同的算法，网络架构和超参数。 这项工作弥合了高维感知输入和动作之间的鸿沟，产生了第一个人工智能体，它能够在多种挑战性任务中，学着变得优秀。（Silver 等，2015）

这里是一个截图，展示了在不同领域中，与线性学习器和人类相比，DQN 的位置。



> 这些按照职业人类游戏测试者来正则化：0% = 随便玩玩，100% = 人类的表现。来源：DeepMind 的 DQN 论文，通过深度强化学习的人类级别控制

为了帮助你构建一些直觉，关于这些进展在 RL 研究中产生，这里是一些改进的例子，关于非线性 Q 函数上的尝试，它可以改善性能和稳定性。

- 经验重放，通过随机化之前的观测值的更长的序列，以及对应的奖励，来避免近期经验的过拟合。这个思路由生物大脑启发：例如老鼠走迷宫，在睡觉期间“重放”神经活动的模式，以便提升迷宫中的未来表现。
- 循环神经网络（RNN）扩展的 DQN。当一个智能体只能看到它的直接环境时（也就是机器老鼠只能看到迷宫的特定区域，而一只鸟可以看到整个迷宫），智能体需要记住更大的地图，以便它记住东西都在哪里。这类似于人类婴儿如何发展出“物体恒存性”（object permanence），来了解东西是存在的，即使它们离开了婴儿的视野范围。RNN 是循环的，也就是，它们允许信息长时间存在。这里是深度循环 Q 网络（DQRN）玩 Doom 的视频，令人印象深刻。

https://medium.com/media/2286543cfd01ba0ac858ada4857dc635?postId=6eacf258b265

> 论文：https://arxiv.org/abs/1609.05521。来源：Arthur Juliani 的“使用 TensorFlow 的简单的强化学习”系列。

2016 年，仅仅在 DQN 论文的一年之后，DeepMind 发布了另一个算法，叫做 Asynchronous Advantage Actor-Critic（A3C），在训练一半的时间之后，超过了 Atari 游戏的最先进的表现（Mnih 等，2016）。A3C 是一种行动-评判算法，组合了我们之前探索的两种方式：它使用行动器（一个决定如何行动的策略网络），以及一个评判器（一个 Q 网络，决定什么是有价值的东西）。Arthur Juliani 写了一个不错的，特别关于 A3C 网络是什么样。A3C 现在是 OpenAI 的 Universe Starter Agent。

从那个时候之后，就有了无数吸引人的突破 -- 从 AI 发明自己的语言，到教会他们自己在多种地形中行走。这个系列仅仅涉及了 RL 前沿的表面，但是我希望它可以作为未来探索的起始点。

另外，我们打算分享这个 DeepMind 智能体学习走路的视频...并且带有声音。拿一些爆米花，打开声音，然后兼证人工智能的所有荣耀。

https://medium.com/media/e7187ecd760a815468c4e79c622dc625?postId=6eacf258b265

> 

练习材料和扩展阅读

代码

- Andrej Karpathy 的 Pong from Pixels 会让你实现并运行第一个强化学习智能体。文章描述了，“我们会学着玩 ATARI 游戏（乒乓），使用 PG，从零开始，来自像素，使用深度神经网络，并且整个东西是 130 行的 Python 代码，仅仅使用 NumPy 作为依赖（Gist 链接）”。
- 下面，我们高度推荐 Arthur Juliani 的 使用 TensorFlow 的简单强化学习教程。它浏览了 DQN，策略学习，行动-评判方法，以及使用 TensorFlow 实现的探索策略。尝试理解它，之后重复实现涉及到的方法。

阅读 + 讲义

- Richard Sutton 的书，Reinforcement Learning: An Introduction（强化学习导论），一本神奇的书，非常值得一读
- John Schulman 的 CS294：深度强化学习（UCB）
- David Silver 的强化学习课程（UCL）

到此为止了！

> 如果你到达了这里，这就是我们希望的所有奖励。
>
> 我们希望你将这个系列看做机器学习的简介。我们在附录中编译了一些我们最喜欢的 ML 资源，如果你准备好来看看兔子洞有多深的话。
>
> 请不要犹豫，向我们提供思路，问题，反馈，或者你最喜欢的 GIF。
>
> 下次再见，
>
> Vishal 和 Samer

> 总结
>
> 这里是一个基础的问题，它启发了这个系列，我们打算也将它给你。
>
> 作为人类，我们的目标函数是什么？我们如何定义，我们在现实生活中使其最大化的奖励？在基本的快乐和痛苦中，我们的奖励定义也倾向于包含混乱的事情，像是正确和错误，满足，爱情，精神，和目的。
>
> 有一些智力领域，在远古时期，它们就致力于“我们的目标函数是什么，或者应该是什么”的问题，它叫做“伦理学”。伦理学的核心问题是：我们应该做什么？我们应该怎么样或者？什么行为是正确或者错误的？答案非常简洁：它取决于你的价值观。
>
> 随着我们创造出越来越多的高级 AI，它会开始远离玩具问题的领域，像是 Atari 游戏，其中“奖励”仅仅由游戏中赢得了多少积分定义。并且它们越来越出现在现实世界。例如自动驾驶，需要使用更复杂的奖励定义做决策。最开始，奖励可能绑定在一些东西上，例如“安全到达目的地”。但是如果强制让它选择，保持原路线并撞击五个行人，还是转向并撞击一个行人，那么它应该不应该转向呢？如果一个行人是孩子，或者持枪的歹徒，或者下一个爱因斯坦呢？这样如何改变决策，以及为什么？如果转向也会毁掉一些值钱的艺术品呢？突然我们有了更加复杂的问题，当我们尝试定义目标函数，并且答案并不简单的时候。
>
> 这个系列中，我们探索了为什么难以对计算机显式规定猫是什么样子 -- 如果你问我们自己是怎么知道的，答案很简单，“直觉” -- 但是我们探索了机器视觉的方式，让计算机自己习得这个直觉。与之类似，在机器道德的领域，可能难以准确规定，如何求解一个行为对于另一个的正确性和错误性，但是，或许机器可以用某种方式习得这些值。这叫做“价值学习问题”，并且它可能是人类需要解决的，最重要的技术问题之一。
>
> 对于这个话题的更多东西，请见 Risks of Artificial Intelligence（人工智能风险）的概要性文章。以及随着你深入到让机器更聪明的世界中，我们鼓励你记住，AI 的进步是个双刃剑，它的两侧都特别锋利。

六、最好的机器学习资源

> 原文：The Best Machine Learning Resources
>
> 作者：Vishal Maini
>
> 译者：飞龙
>
> 协议：CC BY-NC-SA 4.0
>
> 用于制定人工智能、机器学习和深度学习课程表的资源概览。

制定课程表的一般建议

上学获得一个正式学位并不总是可行或者令人满意的。对于那些考虑自学来代替的人，这就是写给你们的。

1. 构建基础，之后专攻兴趣领域

你不能深入每个机器学习话题。有太多药学的东西，并且领域的进展较快。掌握基础概念，之后专注特定兴趣领域的项目 -- 无论是自然语言理解，计算机视觉，深度强化学习，机器人，还是任何其它东西。

2. 围绕最感兴趣的话题设计你的课程表

对于一些长期学习，或事业目标来说，动机远比稍微优化的学习策略重要。如果你玩的开心，你就会进展较快。如果你尝试强迫你自己前进，你就较慢。

> 我们包含了自己探索或高度推荐的资源。这个列表的并不打算非常详尽。有数不清的选项，也有很多选项是没有作用的。但是如果我们错过了不错的资源，它属于这里，请帮助我们。

基础

> 译者注：如果有翻译过来的免费中文版，会以中文版代替。如果中文版收费，则会同时提供英文和中文链接。

编程

- 语法和基础概念：谷歌的 Python 课程、笨办法学 Python。

- 练习：CoderByte、CodeWars、HackerRank。

线性代数

- 深度学习圣经，第二章：线性代数。机器学习相关的线性代数概念的快速概览。
- A First Course in Linear Model Theory（线性模型理论的第一堂课）。Nalini Ravishanker 和 Dipak Dey 所著。在统计学语境下介绍线性代数的课本。

概率统计

- MIT 18.05，概率统计导论，由 Jeremy Orloff 和 Jonathan Bloom 讲授。提供概率归因和统计推断的知道，对于理解机器如何思考、规划、和决策来说，它是无价的。
- All of Statistics: A Concise Course in Statistical Inference（统计大全：统计推断的短期课程），Larry Wasserman 所著。统计学的导论性课本。

微积分

- 可汗学院：微分。或者任何微积分课程或课本。
- 斯坦福 CS231n：导数，反向传播和向量化，Justin Johnson 所著。

机器学习

- 课程
  - 吴恩达的机器学习课程，在网易云课堂上（更严格来说，是斯坦福 CS229。
  - 数据科学训练营：Galvanize（全日制，三个月，较贵）；Thinkful（时间灵活，六个月，较便宜）。
- 课本
  - 统计学习导论（英文，中文），Gareth James 等人所著。本质性机器学习概念的优秀参考，英文版免费。

深度学习

- 课程
  - Deeplearning.ai，吴恩达的导论性深度学习课程。
  - CS231n：用于视觉识别的卷积神经网络，斯坦福的深度学习课程。有助于构建概念，带有吸引人的讲义和演示性的问题集。
- 项目
  - Fast.ai，有趣而遍历的，基于项目的课程。项目包括猫狗的图像分类，和尼采哲学的作品生成。
  - 使用 TensorFlow 分类 MNIST 手写数字。使用这个 Google 的教程，以超过 99% 的

准确率，在三个小时内分类手写数字。
    - 亲自试试 Kaggle 的比赛。实现感兴趣的深度学习论文，使用 Github 上的其它版本作为参考资料。
- 阅读
    - 深度学习圣经，Ian Goodfellow，Yoshua Bengio 和 Aaron Courville 所著。
    - 神经网络和深度学习，清晰和在线刻度的深度学习教程，Michael Nielsen 所著。以一些到达人类级别的智能结尾。
    - 深度学习论文阅读路线图，关键论文的全集，按照时间和研究领域组织。

强化学习

- 课程
    - John Schulman 的 CS294：深度强化学习，位于 UCB。
    - David Silver 的 强化学习课程，位于伦敦大学学院。
    - 深度 RL 训练营，由 OpenAI 和 UCB 组织。应用当前关闭了，但是值得看一看未来的课程。
- 项目
    - Andrej Karpathy 的 Pong from Pixels。从零开始，在 130 行代码之内，实现打乒乓球的智能体。
    - Arthur Juliani 的 Simple Reinforcement Learning with Tensorflow（Tensorflow 简易强化学习）。使用 TensorFlow 实现 Q-learning, policy-learning 和 actor-critic 方法，和探索策略。
    - 更多思路请查看 OpenAI 的 requests for research。
- 阅读
    - Richard Sutton 的书，Reinforcement Learning: An Introduction（强化学习导论）。

人工智能

- 人工智能：一种现代方法（英文，中文），由 Stuart Russell 和 Peter Norvig 所著。
- Sebastian Thrun 的优达学城课程，人工智能导论。
- 奖学金：Insight AI 伙伴计划，谷歌大脑实习项目。

人工智能安全

- 对于短的版本，请阅读：1）Johannes Heidecke 的 Risks of Artificial Intelligence（人工智能的风险）；2）OpenAI 和 谷歌大脑的合作，Concrete Problems in AI Safety（AI 安全中的具体问题）；3）Wait But Why 的文章 AI Revolution（AI 的进化）。
- 对于长的版本，查看 Nick Bostrom 的 Superintelligence（超智能）。
- 查看 MIRI 和 FHI 发布的 AI 安全方面的研究。
- 保持关注 Reddit 上的 `/r/ControlProblem`。

时事通讯

- Import AI，每周的 AI 通讯，涵盖业界的最新发展。由 OpenAI 的 Jack Clark 筹划。
- Machine Learnings，由 Sam DeBrule 筹划。它是这个领域中经常客串的专家。
- Nathan.ai，涵盖近期新闻，和风投视角的 AI/ML 评论。
- The Wild Week in AI，由 Denny Britz 维护。标题说明了一切。

来自其他人的建议

- What is the best way to learn machine learning without taking any online courses?（不上任何在线课程的情况下，什么是学习机器学习的最佳方式？），由谷歌大脑的 Eric Jang 回答。
- What are the best ways to pick up deep learning skills as an engineer?（作为工程师，什么是修炼深度学习技能的最佳方式？），由 OpenAI 的 CTO Greg Brockman 回答。
- A16z 的 AI Playbook（AI 攻略书），更加基于代码的介绍。
- AI safety syllabus（AI 安全大纲），由 80,000 Hours 设计。



> 你选择了蓝色药丸，然后故事结束了。你在你的床上醒来，并且相信了你打算相信的任何东西。你选择了红色药丸，仍然留在仙境中，然后我向你展示兔子洞有多深。-- Morpheus

最后祝你好运，再见。